

**Fachhochschule Aachen, Campus Jülich**

Fachbereich: Medizintechnik und Technomathematik (FB9)  
Studiengang: Technomathematik

**Masterarbeit von Oliver Schmidts**

Matrikelnummer 855387

---

**Verfolgung von Bluetooth-LE Beacons bei fest  
installierten Empfängern**

---

betreut von

Prof. Dr. Bodo Kraft und Dr. Maik Boltes

in Kooperation mit

Forschungszentrum Jülich GmbH  
Jülich Supercomputing Centre  
Abteilung Civil Security and Traffic

Jülich im September 2016



# Eidesstattliche Erklärung

Diese Arbeit ist von mir selbstständig angefertigt und verfasst. Es sind keine anderen als die angegebenen Quellen oder Hilfsmittel genutzt worden.

Oliver Michael Schmidts  
Jülich, 24. September 2016

---

Unterschrift des Studierenden

Diese Arbeit wurde betreut von:

1. Prüfer: Prof. Dr. Bodo Kraft
2. Prüfer: Dr. Maik Boltes



# Kurzzusammenfassung

Im Rahmen dieser Ausarbeitung wird das bisherige Standardszenario der Innenraumlokalisierung mittels Bluetooth Low Energy umgedreht. Das bedeutet, dass anhand von stationären Empfängern untersucht wird, wie gut sich sendende Bluetooth-LE-Beacons durch ein Fingerprinting-Verfahren lokalisieren lassen.

Dazu wird zunächst auf Grundlagen der Bluetooth-Technologie eingegangen, um mögliche Fehlerquellen zwischen Sendern und Empfängern zu identifizieren. Ebenso wird allgemein erläutert, wie Fingerprinting-Verfahren funktionieren und wie das dazu nötige Vorgehen auf die Verwendung einer Support Vector Machine zu übertragen ist.

Mit Hilfe dieser Grundlagen wird das bisherige System zur Ortung von beweglichen Empfängern der Fachhochschule Aachen so erweitert, dass sich auch Beacons lokalisieren lassen. Dazu wird erläutert, wie eine synchrone Steuerung der Empfänger über einen RabbitMQ-Server funktionieren kann und sich die empfangenen Signale in die bisherigen Datenstrukturen integrieren lassen, ohne dass eine Neuimplementation erforderlich wird. Weiterhin wird vorgestellt, wie sich die gemessenen Bluetooth-Signale auf einem Server synchronisieren lassen, sodass eine Ortung über das bestehende System ermöglicht wird.

Aufgrund der Ungenauigkeiten der Positionsbestimmung in einem ersten Anwendungstest des entworfenen Systems, werden verschiedene Möglichkeiten zur Transformation der Rohdaten in einen zur Lokalisierung geeigneten Merkmalsraum einer Support Vector Machine diskutiert. Dabei erweisen sich Verfahren, die mehrere Merkmale verwenden, denen, die einzelnen Merkmale berücksichtigen, deutlich überlegen.

Des Weiteren wird gezeigt, dass Ensemble-Techniken wie ein Mehrheitsvotum, die Lokalisierungsergebnisse des kombinierten Verfahrens unter bestimmten Voraussetzungen übertreffen können. Insgesamt konnten die Resultate aus dem ersten Versuch durch die verschiedenen Optimierungen so verbessert werden, dass verschiedene Anwendungen ermöglicht werden.

Zusätzlich lassen sich alle vorgestellten Verfahren auf bewegliche Empfänger übertragen, sodass alle Optimierungen ebenso auf bisherige Anwendungen genutzt werden können. Dies ermöglicht im klassischen Anwendungsbereich eine Ortungsgenauigkeit auf Raumebene nahezu immer zu erzielen.



# Inhaltsverzeichnis

<b>1</b>	<b>Motivation und Zielsetzung</b>	<b>1</b>
1.1	Verwandte Arbeiten . . . . .	2
1.2	Anwendungsszenarien . . . . .	4
1.2.1	Anwendungsszenario: Fußgängertracking . . . . .	4
1.2.2	Anwendungsszenario: Hybride, mit GPS gekoppelte Systeme . . .	5
1.2.3	Anwendungsszenario: Waren- bzw. Geräteverfolgung . . . . .	5
1.3	Zielsetzung der Arbeit . . . . .	6
<b>2</b>	<b>Grundlagen</b>	<b>9</b>
2.1	Bluetooth 4.0 Low Energy . . . . .	9
2.2	Bluetooth-LE-Beacons . . . . .	10
2.3	Raspberry Pi als stationärer Empfänger . . . . .	13
2.4	Signalgenauigkeit und Störungsanfälligkeit . . . . .	15
2.5	Ortsbestimmung über Fingerprinting . . . . .	18
2.5.1	Allgemeines . . . . .	18
2.5.2	Fingerprinting mit Support Vector Machines . . . . .	19
<b>3</b>	<b>Softwaretechnische Umsetzung</b>	<b>23</b>
3.1	Aufbau des Testsystems . . . . .	23
3.1.1	Ortungsablauf mit beweglichen Empfängern und stationären Sendern . . . . .	23
3.1.2	Data-Objects . . . . .	24
3.1.3	Location-Solver . . . . .	25
3.1.4	Backend-Webservices . . . . .	26
3.2	Integration der Beacon-Ortung . . . . .	29
3.2.1	Synchronisation der Empfänger . . . . .	29
3.2.2	Anpassungen im Backend . . . . .	31
3.2.3	Anwendungstest . . . . .	34
<b>4</b>	<b>Optimierung und Ergebnisse</b>	<b>37</b>
4.1	Konzeption einer Testmethodik . . . . .	37
4.1.1	Vergleichstest für stationär installierte Beacons . . . . .	37
4.1.2	Feldversuche für bewegliche Beacons . . . . .	39
4.1.3	Praxistest für bewegliche Beacons . . . . .	40
4.2	Modelloptimierung . . . . .	41
4.2.1	Bisheriges Modell . . . . .	41
4.2.2	Optimierung . . . . .	43

4.3	Signalanalyse . . . . .	48
4.4	Merkmal-Vektor Optimierung . . . . .	52
4.4.1	Theoretische Überlegungen und Integrationstest . . . . .	52
4.4.2	Praxis und Integrationstest für bewegliche Sender . . . . .	57
4.4.3	Mehrheitsvotum . . . . .	63
4.5	Allgemeine Beobachtungen während der Experimente . . . . .	66
4.5.1	Beobachtungen innerhalb eines Raumes . . . . .	66
4.5.2	Beobachtungen bei Raumwechseln . . . . .	67
<b>5</b>	<b>Zusammenfassung und Fazit</b>	<b>69</b>
5.1	Zusammenfassung . . . . .	69
5.2	Fazit: Grenzen des Systems . . . . .	70
5.3	Fazit: Mögliche Anwendungsgebiete . . . . .	71
<b>6</b>	<b>Ausblick</b>	<b>73</b>
6.1	Vergleichsmöglichkeit und Einsatz mehrerer Klassifikatoren . . . . .	73
6.2	Kombination mit anderen Sensoren . . . . .	74
6.3	Standardisierung von Testszenarien . . . . .	75



# Abbildungsverzeichnis

1.1	Trajektorien mit Kameras ohne Marker . . . . .	5
2.1	Bluetooth-LE Advertisment Kanäle [Arg16] . . . . .	10
2.2	Bluetooth LE Advertisment Paket [War14] . . . . .	10
2.3	Bluetooth LE iBeacon Paket [War14] . . . . .	11
2.4	Platine eines Ubudu uBeacons . . . . .	12
2.5	Raspberry Pi 2 B [Mul16] . . . . .	14
2.6	Ergebnisse einer Messung zum klassischen Szenario . . . . .	17
2.7	Separierung von Klassen durch Hyperebene . . . . .	20
3.1	UML-Diagramm der wichtigsten Datenstrukturen in Data-Objects . . . . .	25
3.2	UML-Diagramm der Datenstrukturen im Location-Solver . . . . .	27
3.3	Grafische Oberfläche der Webanwendung. Abgebildet ist die Visualisierung der Karte im Reiter „Tile Map“ . . . . .	28
3.4	UML-Diagramm der neu hinzugefügten Datenstrukturen zur Kommunikation . . . . .	30
3.5	Grafische Oberfläche zur Beacon-Lokalisierung im gelb umrandeten Reiter „Beacon Localization“ . . . . .	32
3.6	Raumkonfiguration für Feldversuche . . . . .	34
4.1	Visualisierung in der Webanwendung . . . . .	38
4.2	Versuchsaufbau für einen Praxistest . . . . .	41
4.3	Bisheriger Wertebereich des Merkmalraumes . . . . .	44
4.4	Optimierter Wertebereich . . . . .	45
4.5	Ergebnisse des Praxistests . . . . .	47
4.6	Dichte-Histogramm im klassischen Szenario . . . . .	49
4.7	Ergebnisse einer Messung für bewegliche Beacons . . . . .	51
4.8	Merkmalraum über den Median . . . . .	52
4.9	Merkmalraum über 25%-Quantil . . . . .	54
4.10	Merkmalraum über 75%-Quantil . . . . .	55
4.11	Ergebnisse des Praxistests mit Median . . . . .	58
4.12	Ergebnisse des Praxistests mit 25% Quantil . . . . .	59
4.13	Ergebnisse des Praxistests mit 75% Quantil . . . . .	60
4.14	Ergebnisse des Praxistests kombinierten Merkmalen . . . . .	62
4.15	Ergebnisse des Praxistests mit korrelierten Merkmalen im Mehrheitsvotum . . . . .	64
4.16	Ergebnisse des Praxistests mit unkorrelierten Merkmalen im Mehrheitsvotum . . . . .	65



# Tabellenverzeichnis

4.1	Ergebnisse des Integrationstests mit einfachem Mittelwert . . . . .	43
4.2	Ergebnisse des Integrationstests mit gleitendem Mittelwert . . . . .	43
4.3	Ergebnisse mit optimiertem Modell und einfachem Mittelwert . . . . .	46
4.4	Ergebnisse mit optimiertem Modell und gleitendem Fenster . . . . .	46
4.5	Klassifikationsergebnisse mit einfachem Median . . . . .	53
4.6	Klassifikationsergebnisse mit gleitendem Median . . . . .	53
4.7	Testergebnisse mit einfachem 25%-Quantil . . . . .	55
4.8	Testergebnisse mit gleitendem 25%-Quantil . . . . .	55
4.9	Testergebnisse mit einfachem 75%-Quantil . . . . .	56
4.10	Testergebnisse mit gleitendem 75%-Quantil . . . . .	56
4.11	Ergebnisse mit kombinierten Merkmalen . . . . .	57
4.12	Ergebnisse mit gleitenden kombinierten Merkmalen . . . . .	57



# Listings

2.1	Von Noble definierte Schnittstelle bei empfangenen BLE-Signalen . . . .	14
3.1	Über Spring injizierte RabbitMQ-Abhängigkeiten . . . . .	31
3.2	Ein Template für Nachrichten über RabbitMQ. In der Konfiguration wird der Verbindungstyp und die Konvertierer übergeben. . . . .	31
3.3	Publizieren von Nachrichten mit Hilfe eines Templates . . . . .	32
3.4	Empfangen von Nachrichten mit Fingerabdrücken durch Spring-Annotationen im <i>SignalCaptureRestApiController</i> . . . . .	32
3.5	Synchronisierung der Trainingsdaten über relative Zeitstempel . . . . .	33



# 1 Motivation und Zielsetzung

Die Anwendungsmöglichkeiten standortbasierter Dienste reichen von Zusatzinformation zu nahegelegenen Restaurants über Umgebungsinformationen zu Tankstellen, Ärzten oder Krankenhäusern bei Navigationsanwendungen wie Google Maps bis zum automatischen Verschicken des Standortes bei einem schweren Verkehrsunfall (eCall)[Wik16a]. Innerhalb von Gebäuden ist diese Art der Ortung und Navigation bis heute nur sehr eingeschränkt möglich.

Wenn zum Beispiel das Krankenhaus über die Navigation gefunden werden konnte, fällt es danach trotz Beschilderung häufig schwer, sich zu orientieren oder die gewünschte Abteilung zu finden. Über die Innenraumortung oder -lokalisierung kann eine Möglichkeit geschaffen werden, die Navigation innerhalb eines Gebäudes fortzuführen oder zum Beispiel dem behandelnden Arzt bei Betreten eines Zimmers automatisch die richtigen Patientenakten auf ein Tablet zu laden.

Die Idee, in Gebäuden Positionen von Menschen mit Karteninformationen zu verknüpfen, tauchte circa zum Zeitpunkt der Öffnung von GPS für die Öffentlichkeit auf. Allerdings ist eine Ortung über GPS in geschlossenen Räumen nicht möglich, da keine Verbindung zu ausreichend vielen Satelliten zustande kommt und die GPS-Ortung nur eine Genauigkeit von ca. zehn Metern erreicht [Wik16b]. Aus diesem Grund wird vor allem an der Ortung über Funktechnologien wie beispielsweise WLAN, RFID oder Bluetooth geforscht, um eine Innenraumlokalisierung zu erreichen.

Seit ca. 2006 erfreut sich das Gebiet der Innenraumlokalisierung immer größerer Beliebtheit. Dies liegt zum einen an immer besser werdenden Sensoren und ausgefeilteren mathematischen Verfahren. Zum anderen hat die Beliebtheit und Entwicklung von Smartphones und Tablets dafür gesorgt, dass Menschen nahezu ständig Sensoren bei sich tragen. Dadurch wurde Unternehmen die Möglichkeit gegeben, vergleichsweise einfach standortbasierte Dienste zu entwickeln. Mit der Einführung des iBeacon-Standards von Apple steigerte sich die Popularität von positionsabhängigen Anwendungen innerhalb von Gebäuden bei Unternehmen noch einmal deutlich.

Der ursprünglich von Apple geplante Einsatzbereich der iBeacons liegt im Versenden von Werbung bzw. Gutscheinen oder spontanen Angeboten in Supermärkten. Die genaue Ortung einer Person ist für solche Anwendungen nicht nötig, da es ausreichend ist, sich in einem gewissen Radius um das Beacon aufzuhalten. Daher wird nur grob zwischen den Entfernungen unterschieden, zum Beispiel ob sich ein Smartphone eher in der Nähe oder weiter entfernt befindet. Einige weitere zu Angeboten von Firmen, bei denen fest installierte Beacons dazu genutzt werden Informationen oder Werbung an in ihrer Umgebung befindliche Passanten zu leiten, finden sich beispielsweise auf Wikipedia.

Der für die Forschung inzwischen deutlich interessantere Einsatzbereich liegt in der Navigation bzw. Ortung innerhalb von Gebäuden, die durch Beacons vereinfacht wird.

Inzwischen existieren viele Hersteller, die auf Basis dieses Standards versuchen, Anwendungen mit Umgebungskontext anzubieten. Dadurch ist auch die dazu benötigte Standardhardware inzwischen kostengünstig zu erhalten. Aus diesem Grund werden fast ausschließlich Systeme mit fest installierten Sendern und beweglichen Empfängern (Smartphones) eingesetzt und erforscht. Das aktuell prominenteste Beispiel für Anwendungen, bei denen der (GPS-)Standort einer Person eine große Rolle – abgesehen von Navigationsanwendungen wie Google Maps – spielt, ist das Virtual Reality Spiel Pokémon Go.

### 1.1 Verwandte Arbeiten

In der wissenschaftlichen Literatur existieren viele verschiedene Ansätze zur Innenraumortung, die auf unterschiedliche Funktechnologien oder inertielle Sensoren setzen. Die am häufigsten eingesetzten Technologien zur Lokalisierung innerhalb von Gebäuden sind WLAN und Bluetooth, da Bluetooth-Beacons vergleichsweise einfach zu installieren sind und WLAN nahezu überall vorhanden ist.

In dieser Arbeit wird ausschließlich auf Bluetooth 4.0 bzw. Bluetooth Low Energy (BLE) als Technologie eingesetzt. BLE bietet insbesondere nach der Einführung des iBeacon-Standards 2013 durch Apple eine sehr beliebte, günstige und einfache Möglichkeit der Ortung. Im Abschnitt 2.2 wird genauer auf die Funktionsweise von Bluetooth-Beacons eingegangen.

Erste Konzepte zu positionsabhängigen Anwendungen, bei denen die Ortung auf Bluetooth basiert, orientierten sich an der Funktionsweise von GPS. Dabei wurde eine akzeptable Genauigkeit zunächst nur unter Laborbedingungen und mit Spezialhardware erreicht, weil versucht wurde, eine Positionsbestimmung über die unterschiedlichen Signallaufzeiten durchzuführen. Dafür war eine Messgenauigkeit von wenigen Nanosekunden nötig, die nicht erreicht werden konnte. Trotzdem wurde bereits in bestimmten Konfigurationen eine Genauigkeit von ca. einem Meter erzielt [FDW04]. Mit Standardhardware führt eine Trilateration zu einem Fehler von ca. drei Metern [CSE04].

Die aktuell vielversprechendsten Ansätze verwenden aufwändige Fingerprinting-Verfahren, entweder über Machine-Learning oder Bayes-Schätzer, die in der Regel eine Genauigkeit zwischen ca. zwei und vier Metern erreichen [FH14]. Des Weiteren existieren Verfahren, die Triangulation mit Fingerprinting kombinieren. Die erzielte Genauigkeit konnte so auf 2,67 m verbessert werden [SHRB11]. Andere Arbeiten versuchen nicht eine genaue Position zu bestimmen, sondern ein Aufenthaltsgebiet zu einer möglichst hohen Wahrscheinlichkeit zu ermitteln. Das Gebiet kann dabei entweder ein Raum oder ein kleineres Teilgebiet innerhalb eines Raumes (Subraum) sein. Kajioka et al. [KMU<sup>+</sup>14] konnten eine korrekte Lokalisierung auf Raumebene mit bis zu 95% erreichen.

Der Versuchsaufbau verläuft meistens folgendermaßen: Beacons werden an vorher festgelegten Standorten auf einer Etage oder im Gebäude installiert. Dadurch senden die Bluetooth-Beacons kontinuierlich Signale, die von einem Smartphone empfangen werden können. Sobald der Versuch startet läuft eine Person einen festgelegten Weg mit dem Smartphone in der Hand ab. Nachdem die Strecke abgelaufen wurde, ist das Experiment beendet. Dann wird der zurückgelegte Weg durch das zu testende Verfahren bestimmt



und mit dem tatsächlich zurückgelegten Weg verglichen. Bei Echtzeit-Lokalisierungen wird die Positionsbestimmung noch während des Experiments durchgeführt, sodass direkt erkennbar ist, ob ein Verfahren erhebliche Fehler aufweist. Je nach gewähltem Verfahren müssen zuvor Referenzdatensätze, sog. Fingerabdrücke, erstellt werden.

Auch in der Fachhochschule Aachen wurde ein solches Echtzeit-System entwickelt und in einigen Experimenten erprobt. Ziel dieses Softwaresystems ist die Positionsbestimmung eines beweglichen Empfängers auf Subraumbene mit Hilfe von fest in Räumen installierten Beacons. Dabei werden Räume in ca.  $3 \times 3$  Meter große Untergebiete unterteilt. Bei der Ortung wird nicht die genaue Position der Person festgestellt, sondern in welchem Gebiet sie sich aufhält. Zwischen diesen Gebieten lässt sich über eine Android-App zu verschiedenen Punkten navigieren.

Das zur Lokalisierung verwendete Verfahren nutzt einen Fingerprinting-Ansatz. Im Gegensatz zu den häufig verwendeten Bayes-Schätzern oder einem K-Nearest-Neighbour-Algorithmus wird in diesem System auf eine Support Vector Machine gesetzt. In Abschnitt 2.5 wird näher auf die Funktionsweise solcher Verfahren eingegangen.

Szenarien und Anwendungen, bei denen Beacons an festen Positionen installiert sind und sich ein Smartphone durch einen Gebäudeabschnitt bewegt, sind inzwischen schon intensiv erforscht worden. In dieser Arbeit soll der Aufbau umgedreht werden. Ziel ist, Empfänger an festen Positionen zu installieren und die Beacons zu orten. Diese sollen allerdings nicht stationär im Gebäude verteilt sein, sondern sich bewegen können. Für Versuche bedeutet das: Ein Mensch läuft mit einem Bluetooth-Beacon anstelle eines Smartphones eine Strecke durch einen Raum oder ein Gebäude ab.

Während des Experiments müssen die Signale von mehreren fest installierten Empfängern aufgenommen werden. Danach muss eine Synchronisation der Signale erfolgen, bevor ein zentraler Server die Position der Versuchsperson bestimmen kann. Darauf können die ermittelten Positionen – wie im klassischen Anwendungsfall – mit den tatsächlichen verglichen werden.

Bewegliche Beacons zur Ortung zu verwenden, hat im Vergleich zu den stationären Beacons vor allem einen preislichen Vorteil, wenn eine größere Menge an Objekten oder Personen verfolgt bzw. navigiert werden soll. Außerdem lassen sich beliebige Objekte mit der Beacon-Technologie versehen, da der reine Chip sehr klein ist und die Größe eines Beacons in der Regel nur durch die Größe des Akkus bestimmt wird. Weiterhin kann mit einer entsprechenden App auch ein Smartphone in ein Beacon verwandelt werden. In diesem Fall ist das Smartphone nicht mehr der Empfänger der Bluetooth-Signale, sondern der Sender. Das heißt, alle bisherigen Anwendungsfälle lassen sich mit beweglichen Sendern und statischen Empfängern ebenfalls realisieren. Lediglich der Installationsaufwand würde dadurch gesteigert.

## 1.2 Anwendungsszenarien

### 1.2.1 Anwendungsszenario: Fußgängertracking

Ein Forschungsschwerpunkt der Abteilung Civil Security and Traffic des Jülich Supercomputing Centre ist die Modellierung von Fußgängerdynamiken und Entfluchtungssimulationen. Unter anderem werden Entscheidungen zur Wegfindung durch eine Raumgeometrie von Fußgängern, kräftebasierte und kollisionsfreie Bewegungsmodelle mit Hilfe des Open-Source Simulationsframeworks JuPedSim [JuP16] untersucht. Auch lassen sich in diesem Framework äußere Einflüsse, wie zum Beispiel Feuer und Rauchentwicklung, auf die Wegfindung der Fußgänger simulieren. Damit sichergestellt ist, dass die Entfluchtungssimulation realitätsgetreu ist, existiert eine Richtlinie für mikroskopische Entfluchtungs-Analysen (RIMEA [RIM16]), in der verschiedene Szenarien mit geforderten Ergebnissen zu finden sind.

Zusätzlich werden zur Kalibrierung und Validierung der Simulationsmodelle umfangreiche Laborexperimente, wie im BaSiGo-Projekt [BaS15], durchgeführt. Dazu ist es von großer Wichtigkeit, die räumlichen Bewegungen der Fußgänger während aller Experimente und Feldversuche so genau wie möglich zu verfolgen, das heißt Trajektorien möglichst jedes einzelnen Fußgängers zu bestimmen, um daraus Schlüsse ziehen zu können und die Modelle gegebenenfalls anzupassen.

Bisher geschieht das Extrahieren der Trajektorien mit Hilfe von Kameras, mit denen die Laborexperimente aufgenommen werden, bis auf wenige Zentimeter genau. Dazu müssen Personen spezielle Marker, häufig in Form von Hüten, tragen. In Abbildung 1.1 werden die Teilnehmer eines Experiments an einer drei Meter breiten Ecke ohne Marker über die Kamerabilder lokalisiert, sowie die bisherige Bewegung aufgezeichnet. Teil a) zeigt die Bewegung der Teilnehmer in der letzten Sekunde. In Abbildung 1.1b sind dagegen alle Trajektorien des Experiments eingezeichnet.

Für diese Art der Personenverfolgung ist es allerdings notwendig, dass die Versuchspersonen keine gravierenden Größenunterschiede haben. Auch die Verdeckung von einer Person durch eine andere über verschiedene Frames bereitet Probleme bei der Extraktion von Trajektorien. Deshalb wird nach weiteren Möglichkeiten wie zum Beispiel inertielle Sensoren oder Funktechnologien gesucht, um Menschen in Laborexperimenten oder Feldversuchen möglichst genau verfolgen zu können.

Bisherige Ergebnisse zeigen, dass beim Einsatz von Bluetooth maximal eine Genauigkeit von zwei bis drei Metern erzielt werden kann. In einigen Fällen, wie zum Beispiel Raumwechselentscheidungen oder Versuche mit einer geringen Personendichte, ist die Genauigkeit der Technologie jedoch ausreichend, da Räume bisher mit bis zu 95% detektiert werden können. Dadurch entsteht eine kostengünstige Möglichkeit, Experimente zur Wegfindung von Fußgängern in einem Räumungsszenario durchzuführen. Die Trajektorien bestehen in diesem Fall allerdings nur aus Bewegungen zwischen Räumen und beschreiben nicht den kompletten zurückgelegten Weg, wie es bei der Verfolgung durch Kameras der Fall wäre.

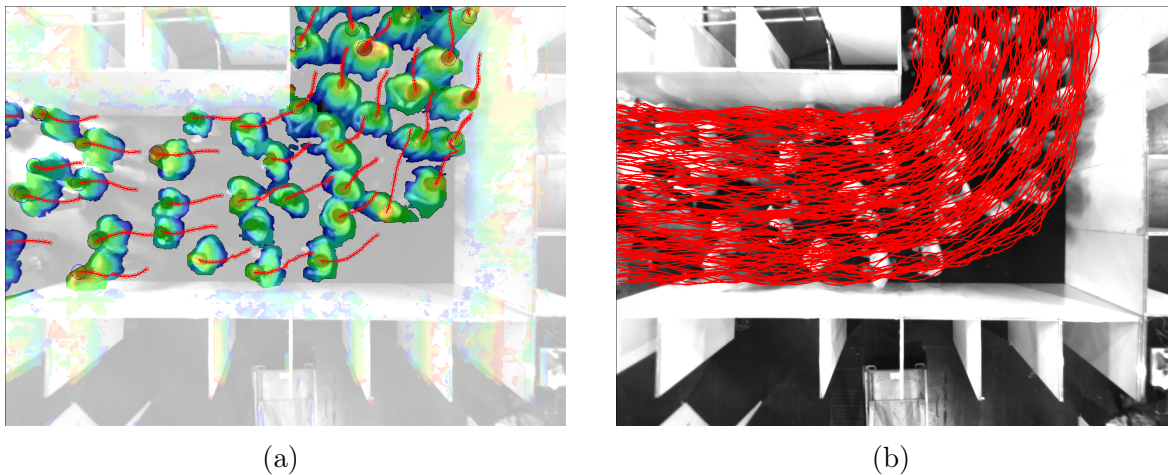


Abbildung 1.1: Trajektorien aus einem Experiment an einer 3m breiten Ecke, bei dem die Personen keine Marker tragen. Rote Linien in a) sind die Laufwege der Personen in der letzten Sekunde. b) zeigt alle 162 Laufwege des Versuchs.[Bol15]

### 1.2.2 Anwendungsszenario: Hybride, mit GPS gekoppelte Systeme

Großwarenhäuser werden von vielen Unternehmen wie zum Beispiel Amazon immer häufiger vollautomatisch betrieben. Das heißt, die Waren werden von Robotern aus den Regalen geholt und später von Menschen verpackt. Allerdings kann nicht jedes Warenhaus vollautomatisiert werden und nicht jede Firma sich einen solchen Umstieg leisten. Eine weitere mögliche Anwendung beweglicher Beacons ist deshalb eine Navigation bzw. Ortung der Gabelstaplerfahrer, um ihre Wege möglichst effizient zu gestalten. Bei dieser Anwendung bietet es sich an, ein Smartphone als beweglichen Sender zu benutzen und Empfänger zu installieren, die dauerhaft an eine Stromversorgung angeschlossen sind. Dadurch ließe sich ein wartungsarmes System installieren.

Eine weitere Möglichkeit zur Anwendung eines solchen System ist die Navigation der Fahrer über größere Firmengelände. Soll beispielsweise ein Paket aus Warenhaus A ins Warenhaus B gebracht werden, kann eine Ortung der Gabelstapler außerhalb eines Warenhauses mittels GPS erfolgen. Allerdings muss der Fahrer in beiden Lagerhäusern die richtigen Regale finden. Dazu wird eine Innenraumlokalisation benötigt, die eine Navigation innerhalb der Warenhäuser ermöglicht. Eine solche Hybridlösung könnte auch außerhalb einer reinen Warenverwaltung erfolgen, wenn eine Navigation über mehrere Gebäude notwendig ist.

### 1.2.3 Anwendungsszenario: Waren- bzw. Geräteverfolgung

Da einfache Beacons die ungefähre Größe eines Daumennagels besitzen können, lassen sie sich leicht in verschiedene Systeme integrieren. Bei der Warenverfolgung wäre beispielsweise denkbar, dass ein Beacon auf ein Paket geklebt wird. Damit ließe sich das Paket

vom Warenhaus ohne Personalaufwand automatisch bei Ankunft in einem Verteilerzentrum registrieren, ohne dass ein Barcode oder ähnliches eingescannt werden müsste. Dadurch wäre technisch sichergestellt, dass sich Pakete zu einem bestimmten Zeitpunkt in dem registrierten Bereich aufgehalten haben. Dass Mitarbeiter vergessen ein Paket zu scannen würde dadurch ausgeschlossen. Nach der Ankunft müsste das Personal die Pakete nur noch auf die richtigen Fließbänder legen und ein computergesteuertes System könnte die Weiterleitung übernehmen. Hierbei würde eine Ortungsgenauigkeit auf Raumebene völlig ausreichen. Durch ein solches System könnte der Verlust von Paketen minimiert werden, da eindeutig festzustellen ist, auf welchem Weg das Paket verloren wurde.

Des Weiteren ließen sich Geräte mit aufgeklebten Beacons beliebig innerhalb eines Gebäudes verfolgen. Eine Anwendung reicht dabei von „Gerät A hat einen Raum verlassen oder betreten“ bis hin zu einem genauen Aufenthaltsprotokoll des Geräts über den gesamten Tag. Dadurch könnte auch die Aufenthaltsdauer pro Raum protokolliert werden. Dies ist zum Beispiel sinnvoll, wenn bewegliche Geräte nur in bestimmten Räumen oder Gebäudebereichen oder innerhalb eines Gebiets nur für eine bestimmte Zeit genutzt werden dürfen. Ein solches System würde auch in geringem Maße vor einem Verlust der Geräte durch Verlegen schützen.

### 1.3 Zielsetzung der Arbeit

Innerhalb dieser Arbeit soll das System der Fachhochschule Aachen zur Lokalisierung von beweglichen Empfängern mit stationären Bluetooth 4.0 Sendern um die Fähigkeit erweitert werden, die Positionsbestimmung auch im umgekehrten Szenario mit beweglichen Sendern und stationären Empfängern durchzuführen. Dabei wird ausschließlich auf dem Markt erhältliche günstige Standardhardware verwendet, um mögliche Anwendungen vor allem im Bereich der Fußgängerdynamik günstig zu halten. Deshalb werden alle Feldversuche innerhalb dieser Arbeit mit mehreren Raspberry Pi 2 B als Empfänger und mit uBeacons der Firma Ubudu als Sender durchgeführt. Die Experimente mit beweglichen Sendern finden im Raum 01c04 der Fachhochschule statt. Da das vorhandene System erweitert werden soll, wird zur Lokalisierung das Fingerprinting-Verfahren der Fachhochschule genutzt.

Der erste Teil der Arbeit widmet sich den technischen Anforderungen einer solchen Erweiterung, während der zweite Teil auf die Verbesserung der Resultate der ersten Funktionstests abzielt. Die Basisanforderung an diese Arbeit ist herauszufinden, ob ein Szenario mit beweglichen Sendern in der Praxis grundsätzlich funktionieren kann. Dabei muss insbesondere darauf geachtet werden, dass eine Live-Lokalisierung und -Verfolgung umgesetzt wird. Das heißt, die Daten aus Personenbewegungen werden nicht aufgenommen und im Nachhinein auf ihren Standort analysiert, sondern, während sich eine Person mit einem Sender durch einen Raum bewegt, muss die Position möglichst in Echtzeit bestimmt werden. Um dies zu gewährleisten, müssen verschiedene Probleme wie die Synchronisation der Empfänger und die Einbindung der empfangenen Signale in das aktuelle System gelöst werden.

Neben diesen Anforderungen zur rein technischen Umsetzung muss zur Beurteilung der Machbarkeit einer Verfolgung von Bluetooth-LE-Beacons mit fest installierten Empfängern die mögliche Genauigkeit untersucht werden. Diese Untersuchungen zielen nicht nur auf die Verbesserung dieses Anwendungsfalls ab, sondern sollen sich gleichzeitig direkt auf das bisherige Standardszenario mit beweglichen Empfängern anwenden lassen. Dadurch können beide Ortungsverfahren verbessert werden. Dazu müssen verschiedene Untersuchungen zu dem bisherigen Fingerprinting-Verfahren und eine Optimierung durchgeführt werden.

Da diese Arbeit in Kooperation mit der Abteilung Civil Security and Traffic des Forschungszentrums Jülich entsteht, ist nach einer Untersuchung der möglichen Genauigkeit dieses Verfahrens ein besonderes Augenmerk auf die Anwendbarkeit zur experimentellen Fußgängerverfolgung zu legen. Aus diesem Grund muss die Genauigkeit der Ortung mindestens auf Raumebene funktionieren, um sinnvolle experimentelle Anwendungen zu ermöglichen. Daneben sollen Grenzen dieser Technologie aufgezeigt werden. Dazu gehören die Untersuchung verschiedener Einflüsse auf die Ortungsqualität und die Robustheit der Positionsbestimmung.



## 2 Grundlagen

In diesem Kapitel werden die Grundlagen für das Verständnis der im weiteren Verlauf genutzten Techniken erörtert. Dabei wird zunächst Bluetooth als verwendete Technologie allgemein erläutert. Außerdem wird auf die Funktionsweise der Beacons eingegangen und wie Signale von einem Raspberry Pi als Empfänger aufgenommen werden können. Des Weiteren werden die verschiedenen Ursachen von Signal-Störungen dargestellt. Weiterhin wird die grundsätzliche Funktionsweise eines Fingerprinting-Verfahrens zur Ortsbestimmung vorgestellt.

### 2.1 Bluetooth 4.0 Low Energy

Bluetooth ist ein frei verfügbarer Industrie-Standard zum drahtlosen Austausch von Daten, der seit 1994 ständig von der Bluetooth Special Interest Group (SIG) weiterentwickelt wird [SIG16]. Bluetooth funkt, genau wie WLAN, auf dem 2,4 GHz ISM (Industrial, Scientific and Medical) Band. Bei Bluetooth 4.0 wird dieses Band, wie in Abbildung 2.1 zu sehen ist, in 40 Kanäle (Channel) eingeteilt, wobei sich ein Teil der Kanäle mit WLAN-Kanälen überschneidet.

Da klassisches Bluetooth – die Versionen vor Bluetooth 4.0 – 79 Kanäle verwendet, ist eine Abwärtskompatibilität seitens des Standards nicht mehr vorhanden. Die drei gelb-orange markierten Kanäle sind für Broadcasting reserviert und werden Advertisement Channels genannt. Auf diesen Kanälen ist kein Verbindungsaufbau zur Datenübertragung notwendig. Dieser Teil des Standards liefert dadurch die Grundlage für werbeverteilende iBeacons und somit auch zur Innenraum-Lokalisierung.

Neben der Möglichkeit, Broadcasting Pakete zu versenden, ist Bluetooth 4.0 sehr energiesparend, sodass eine Laufzeit von mehreren Monaten mit einer einzigen Knopfzelle ermöglicht wird [SIG16].

Der Aufbau der Bluetooth-Advertisement-Pakete ist in Abbildung 2.2 dargestellt. Die Zugriffsadresse (Access Address) ist für alle Advertisement Pakete identisch, da kein Verbindungsaufbau erfolgen muss. Innerhalb der PDU (Protocol Data Unit) können 31 Bytes frei mit Daten gefüllt werden. Dies könnten anstelle der für die Werbung relevanten iBeacon-Pakete auch verschiedene Sensordaten wie zum Beispiel Temperatur oder Luftfeuchtigkeit sein. Da weitere Details aus dem Bluetooth 4.0 Standard für die Positionsbestimmung mit Bluetooth-Beacons keine entscheidende Rolle spielen, wird darauf an dieser Stelle nicht weiter eingegangen.

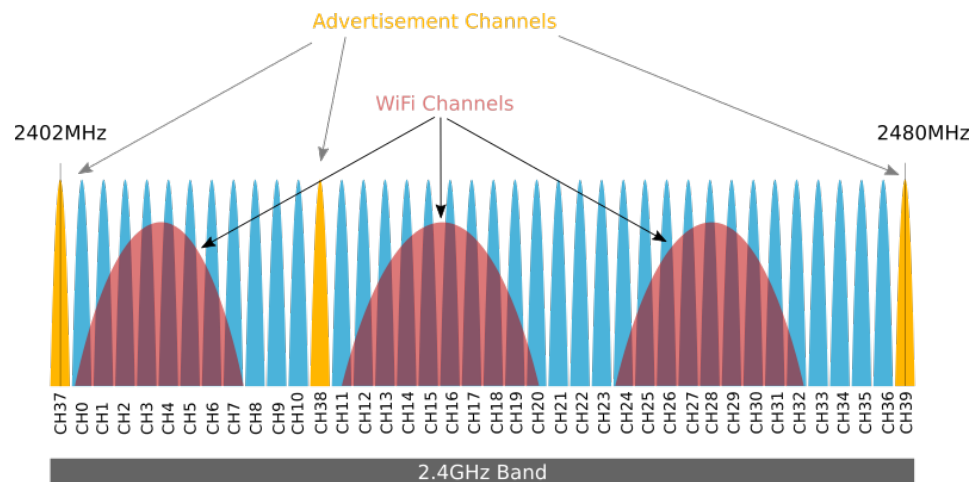


Abbildung 2.1: Bluetooth-LE Advertisement Kanäle [Arg16]

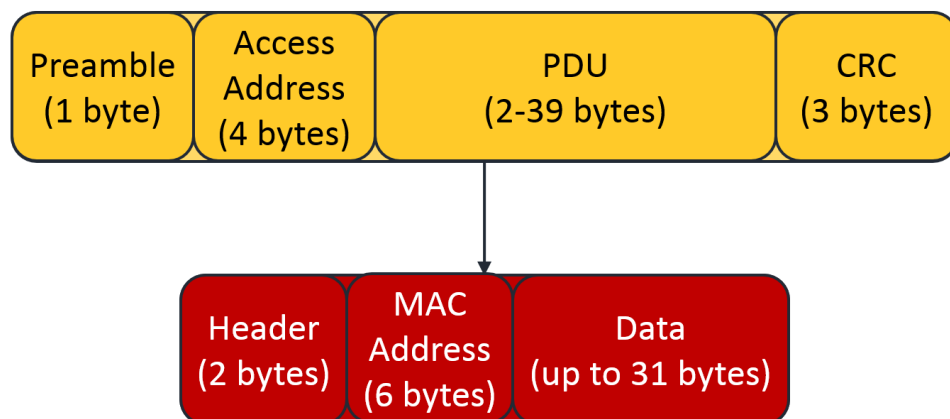


Abbildung 2.2: Bluetooth LE Advertisement Paket [War14]

## 2.2 Bluetooth-LE-Beacons

Bluetooth-LE-Beacons wurden, wie beschrieben, mit dem iBeacon-Standard von Apple eingeführt und erfreuen sich immer größerer Popularität. Das Wort „beacon“ steht im Englischen für Leuchtfener. Auf diese Weise verhält sich auch ein Bluetooth-Beacon: Es broadcastet ununterbrochen Signale in die entsprechenden Bluetooth-Kanäle, bis es abgeschaltet wird oder die Batterie versagt.

Apples ursprüngliches Ziel war, das Versenden von Werbung oder Gutscheinen in Einkaufszentren. Dazu war eine einfache Innenraum-Lokalisierung nötig. Diese kann ausschließlich die Typen direkt (*immediate*), nah (*near*), fern (*far*) und unbekannt (*unknown*) in einem Radius um ein einzelnen Beacon unterscheiden, liefert jedoch keine genaue Entfernung. Laut Spezifikation bedeutet *near*, dass ein Beacon in einem Umkreis von einem bis zu drei Metern zu finden ist [App14].

Für ortsbezogene Anwendungen, die lediglich eine ungefähre Entfernung benötigen, ist diese Genauigkeit völlig ausreichend. Um jedoch Personen oder Gegenstände ver-



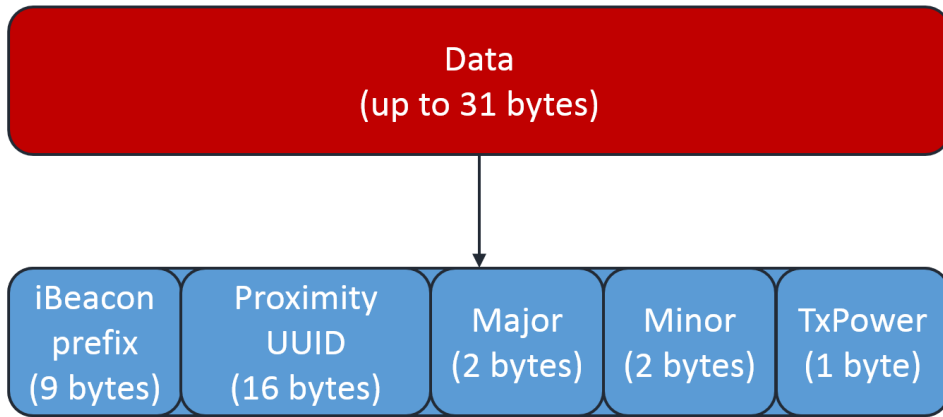


Abbildung 2.3: Bluetooth LE iBeacon Paket [War14]

folgen und navigieren zu können, wird eine deutlich höhere Genauigkeit benötigt als „in der Nähe“. Viele Hersteller von Bluetooth-Beacons haben ihre Produkte von Apple lizenzieren lassen, sodass viele Hersteller für Bluetooth-Beacons existieren, die den iBeacons-Standard unterstützen.

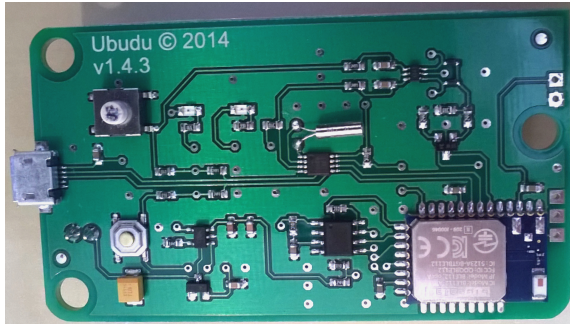
Abbildung 2.3 zeigt den Aufbau des Pakets innerhalb der PDU des Advertisement-Pakets. Dabei ist das iBeacon Präfix ein fester Wert für alle iBeacons. Über den Inhalt der Felder *Proximity UUID*, *Major* und *Minor* lässt sich ein iBeacon identifizieren und im Geschäftsalltag einer spezifischen Anwendung zuordnen. Eine Identifizierung des iBeacons kann ebenfalls über die MAC-Adresse in der PDU erfolgen. Daher werden alle anderen Felder, über die sich Beacons identifizieren lassen, nicht benötigt.

Das für eine Lokalisierung entscheidende Feld ist *TxPower*. Das Zweierkomplement der *TxPower* liefert einen Referenzwert (Referenz-RSSI), wie stark das Signal in einem Meter Entfernung sein sollte. Die Bluetooth 4.0 API liefert dazu einen Indikator zur empfangenen Feldstärke im Bluetooth-Empfänger (Received Signal Strength Indicator, RSSI in dBm). Aus diesen beiden Werten und einer einfachen Signaldämpfung lässt sich mit Hilfe der Formeln 2.1 und 2.2 bereits eine Distanz ableiten [Tre15].

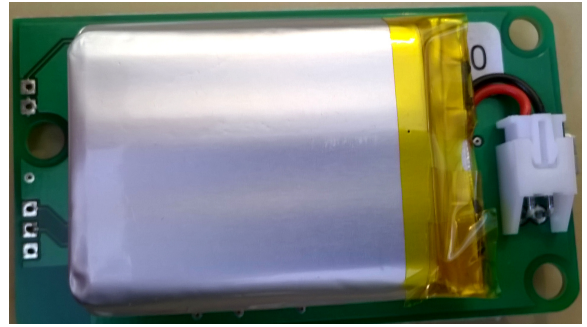
$$RSSI = -10 \cdot \log_{10}(Distanz) - TxPower \quad (2.1)$$

$$Distanz = 10^{\frac{-RSSI - TxPower}{10 \cdot n}} \quad (2.2)$$

Die RSSI-Werte bilden die Grundlage der Distanzberechnung, da *TxPower* in jedem Beacon fest eingestellt ist. Ein höherer RSSI-Wert spricht für eine geringere Entfernung und anders herum. Generell lässt sich sagen, dass der vom Empfänger gemessene RSSI-Wert im Bereich von  $-30$  bis  $-100$  dBm liegt. Auf die direkte Ermittlung der Distanz mit Formel 2.2 sollte allerdings verzichtet werden, da sie anfällig für Messfehler und Störungen ist. Deshalb ist die so ermittelte Distanz in der Praxis nicht aussagekräftig. Selbst unter Laborbedingungen beeinflussen Messfehler die einfache Distanzberechnung deutlich [Tre15]. Im Abschnitt 2.4 wird genauer auf mögliche Probleme beim Signalempfang bzw. Störungen des Signals eingegangen.



(a) Platine eines Ubudu uBeacons von vorne



(b) Platine eines Ubudu uBeacons von hinten

Abbildung 2.4: Platine eines Ubudu uBeacons

Für Experimente, die im Rahmen dieser Arbeit durchgeführt werden, wird auf Bluetooth-LE-Beacons des Herstellers Ubudu zurückgegriffen. Abbildung 2.4 zeigt die Platine eines uBeacons ohne Schutzhülle. Dabei ist deutlich zu erkennen, dass der Akku die Größe der Platine bestimmt. Der eigentliche Beacon-Chip besitzt lediglich die Größe eines Daumennagels.

uBeacons erfüllen den iBeacon-Standard und lassen sich in gewissen Grenzen konfigurieren. Die maximal mögliche Sendefrequenz liegt bei 10 Hz. Allerdings haben Versuche gezeigt, dass es sich hierbei nur um einen Richtwert und keine zuverlässige Angabe handelt, da die Sendefrequenz mit dem Ladestand des Akkus korreliert ist. Je geringer dieser ist, desto niedriger wird auch die Sendefrequenz der Advertisement-Pakete.

Zusätzlich unterliegt das Sendeintervall weiteren Ungenauigkeiten, sodass 10 Hz in den seltensten Fällen tatsächlich erreicht werden. Die *TxPower* ist bei allen Beacons so eingestellt, dass die Referenz RSSI – ermittelt über das Zweierkomplement – bei  $-50$  dBm liegt.

Neben den im Einsatz befindlichen uBeacons können auch andere Geräte mit entsprechender Hard- und Software zu Beacons umfunktioniert werden. Dies betrifft zum Beispiel jedes Smartphone oder Tablet mit Bluetooth 4.0 fähiger Hardware und Android (ab Version 5.0) [You13]. Statt Beacons, die nur Signale senden, könnten in Zukunft daher auch „smarte“ Beacons verwendet werden, indem das Smartphone über eine entsprechende App als solches fungiert.

Dadurch wären auch Anwendungen mit beweglichen Sendern und fest installierten Empfängern auf dem Smartphone denkbar. Ein Vorteil dieser Anwendung wäre, dass Rohdaten, wie beim Szenario mit beweglichen Empfängern, direkt an ein Smartphone gekoppelt würden. Dies ist bei beweglichen Empfängern mit einem modernen Android-Systemen (5.0 und höher) nicht möglich, da mehrere Pakete vom gleichen Sender herausgefiltert und nicht angezeigt werden, wenn sie innerhalb von ca. einer Sekunde empfangen werden. Dies liegt an den ab Android 5.0 neuen Energiesparoptionen [The14]. Dadurch ist eine genaue Innenraum-Lokalisierung ausgeschlossen.

## 2.3 Raspberry Pi als stationärer Empfänger

Bei dem Raspberry Pi handelt es sich um einen Einplatinencomputer, der von der Raspberry Pi Foundation ursprünglich für die Lehre und zum Experimentieren mit Hard- und Software entwickelt wurde. Inzwischen ist dieses Projekt mehrfach ausgezeichnet worden. Stand September 2016 wurden innerhalb von vier Jahren über zehn Millionen Exemplare verkauft. Raspberry Pis existieren in verschiedenen Ausstattungsvarianten und kosten ca. 30 bis 40 Euro [Fou16]. In dieser Arbeit wird ein Raspberry Pi 2 B, wie in Abbildung 2.5 dargestellt, eingesetzt.

Die technische Ausstattung dieser Ausführung setzt sich aus

- 1 GB RAM,
- einer 900 MHz quad-core ARM Cortex-A7 CPU,
- vier USB 2.0 Anschlüssen,
- 40 GPIO Pins,
- einem HDMI-Anschluss,
- einem Ethernet-Anschluss,
- einem 3,5 mm Klinke-Anschluss,
- einem SD-Karten-Einschub,
- sowie mehreren Grafikeinheiten

zusammen.

Nicht enthalten in der Standardkonfiguration sind WLAN und Bluetooth. Da diese Schnittstellen jedoch für den Einsatz der Raspberry Pis als Empfänger benötigt werden, müssen sie über entsprechende USB-Adapter nachgerüstet werden. Das Betriebssystem wird auf einer SD-Karte installiert. Für diese Ausarbeitung wird Ubuntu Mate 16.04 genutzt. Danach kann über die NodeJS-Bibliothek *Noble* [Mis13] direkt auf alle eingehenden Bluetooth-Signale reagiert werden.

Über die *peripheral*-Schnittstelle von Noble, lassen sich die Daten eines jeden Pakets auslesen. Für eine reine Anwendung zur Lokalisierung werden lediglich die Bluetooth-Mac-Adresse des Beacons zur Identifizierung und die empfangene RSSI benötigt. Diese werden von Noble in einem *peripheral*-Objekt gespeichert, das wie in Listing 2.1 aufgebaut ist. Über die gelb markierten Felder *address* und *rss* sind die benötigten Informationen auslesbar. Alle weiteren im Objekt gespeicherten Informationen werden nicht benötigt.

Für spätere Anwendungen könnten sie allerdings Verwendung finden, wenn zum Beispiel mit Smartphones als Beacons gearbeitet werden soll. Dann könnten vor allem die im *Advertisement* gespeicherten Daten Bedeutung erlangen, da sie während der Laufzeit modifizierbar wären.

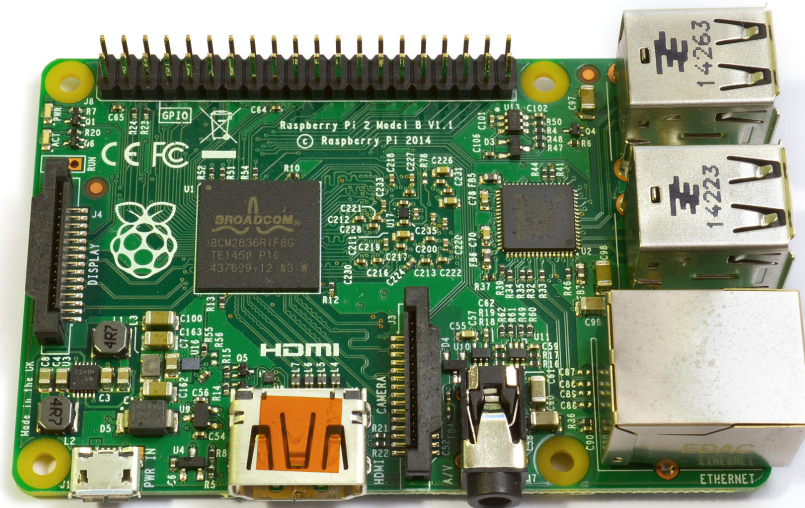


Abbildung 2.5: Raspberry Pi 2 B [Mul16]

Listing 2.1: Von Noble definierte Schnittstelle bei empfangenen BLE-Signalen

```

1 peripheral = {
2   id: "<id>",
3   address: "<BT address>", // Bluetooth Address of device, or "unknown" if not known
4   addressType: "<BT address type>", // Bluetooth Address type (public, random), or ←
      "unknown" if not known
5   connectable: <connectable>, // true or false, or undefined if not known
6   advertisement: {
7     localName: "<name>",
8     txPowerLevel: <int>,
9     serviceUuids: ["<service UUID>", ...],
10    serviceSolicitationUuid: ["<service solicitation UUID>", ...],
11    manufacturerData: <Buffer>,
12    serviceData: [
13      {
14        uuid: "<service UUID>"
15        data: <Buffer>
16      },
17      ...
18    ]
19  },
20   rssi: <rssi>
21 };
22
23 noble.on("discover", callback(peripheral));

```

## 2.4 Signalgenauigkeit und Störungsanfälligkeit

Nachdem geklärt wurde, wie Beacons als Sender und Raspberry Pis als Empfänger funktionieren, muss ein Blick auf Faktoren geworfen werden, die Messergebnisse verfälschen bzw. beeinträchtigen können. Dazu muss zunächst ausgeschlossen werden, dass durch eine mangelnde Abtastrate Aliasing-Effekte auftreten.

Die eingesetzten Beacons senden ihre Signale mit einer Frequenz von 10 Hz. Das heißt, ein Raspberry Pi muss alle 100 ms ein Signal empfangen können. In einem kurzen Test mit mehreren Beacons zeigte sich, dass der Bluetooth-Adapter des Raspberrys mindestens alle 2 ms ein Signal aufnehmen kann. Die Abtastrate ist daher deutlich höher als die nötige Nyquist-Frequenz von 20 Hz.

Sollte ein Signal nicht empfangen werden, kann deshalb davon ausgegangen werden, dass die Signalstärke zu schwach war, um es zu empfangen und der Verlust nicht mit einer zu niedrigen Abtastrate begründet werden kann. Tatsächlich kann davon ausgegangen werden, dass die Abtastrate höher ist als 50 Hz.

Abgesehen von Messungenauigkeiten können verschieden Einflüsse auf ein Funksignal wirken und somit das eigentliche Signal verfälschen. Innerhalb von Gebäuden stellen diese Einflüsse eine besondere Herausforderung dar, da hier mehr Einflüsse auf ein Funksignal wirken als auf einem freien Feld. Dies gilt insbesondere für RSSI Werte im Bluetooth- oder WLAN-Bereich.

Viele Menschen kennen das Problem einer schlechten WLAN- oder Funkverbindung aus dem Alltag, ebenso wie den plötzlichen Verlust einer WLAN-Verbindung. Da WLAN und Bluetooth im gleichen Frequenzbereich senden, wie in Abbildung 2.1 zu erkennen ist, lassen sich diese Erfahrungen auf Bluetooth-Signale übertragen. Die Gründe für plötzliche Signalschwankungen oder gar das vollständige Verschwinden eines Signals können sehr vielfältig sein:

- Messfehler,
- Hindernisse, u.a. Menschen, im Raum,
- Wände,
- Temperaturschwankungen,
- Winkel von Empfänger zum Sender.

Während der Einfluss von Temperaturschwankungen bisher, abgesehen davon, dass er vorhanden ist, nicht genauer untersucht wurden, sind zu den anderen Einflüssen bereits verschiedene Experimente und Studien durchgeführt worden. So kommen Fahrager et. al. [FH14] zu dem Schluss, dass Wände zu einem Verlust von 2 bis 3 dBm führen.

Eine Person in der direkten Sichtlinie von Sender und Empfänger wirkt sich mit einem Signalstärkeverlust von bis zu 20 dBm aus. Dies liegt an einer schlechten Durchdringbarkeit von Wasser durch elektromagnetische Wellen sowie dem Aufbau des menschlichen Körpers, der bei Erwachsenen zu ca. 70% aus Wasser besteht. Vorherige Untersuchungen von Kaemarungsi und Krishnamurthy [KK04] zeigen, dass die reine Anwesenheit einer

Person im gleichen Raum ausreicht, um das Signal eindeutig zu beeinflussen. Vor allem die Varianz der RSSI-Werte wird dabei deutlich verstärkt.

Während die Dämpfung der Signalstärke durch Wände als etwas Positives bezeichnet werden kann, da Wände in der Regel Räume begrenzen und so die Lokalisierung vereinfachen, könnte sich der Einfluss von Menschen auf das Signal insbesondere zur Fußgängerverfolgung als problematisch erweisen. In späteren Experimenten muss aus diesem Grund auf diesen Einfluss geachtet werden. Zusätzlich dazu können Hindernisse im Raum das Signal ähnlich wie Wände beeinflussen. Allerdings sind sowohl Wände als auch Hindernisse für einen Ort im Gebäude als signifikant zu betrachten, weshalb das Signal trotz Verfälschung zur Positionsbestimmung genutzt werden kann.

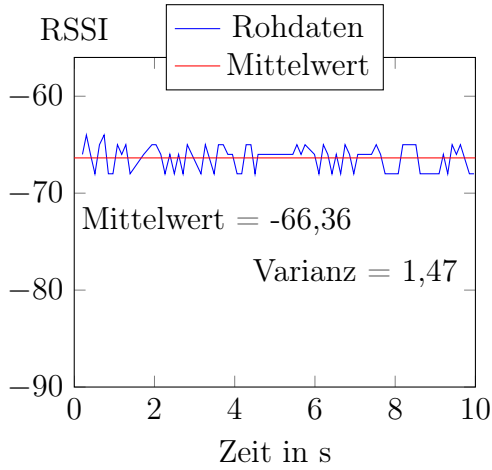
Ein weiterer zu berücksichtigender Einfluss auf ein Signal ist die Ausrichtung vom Sender zum Empfänger. In der Anwendung von stationären Sendern und beweglichen Empfängern, kann dieser weitestgehend vernachlässigt werden [Tre15]. Im Gegensatz dazu ist zu erwarten, dass die Ausrichtung eines beweglichen Senders die Qualität der Signale stark beeinflusst, da der Sender sich mit einer Person dreht und dadurch andere Hindernisse bzw. Einflüsse auf das Signal auf dem Weg vom Sender zum Empfänger einwirken.

Aus diesem Grund wurde ein Versuch durchgeführt, bei dem ein Beacon als Sender in verschiedenen Abständen (1 m, 2 m und 9,5 m) zu einem Raspberry Pi aufgebaut wurde. Die Messdauer während des Versuchs betrug zehn Sekunden. Abbildung 2.6 zeigt die Signalverläufe in den verschiedenen Abständen, wobei sich in der linken Spalte immer das direkt vom Sender auf den Empfänger gerichtete Signal findet. Auf der rechten Seite ist das empfangene Signal des um 90 Grad gedrehten Senders zu sehen.

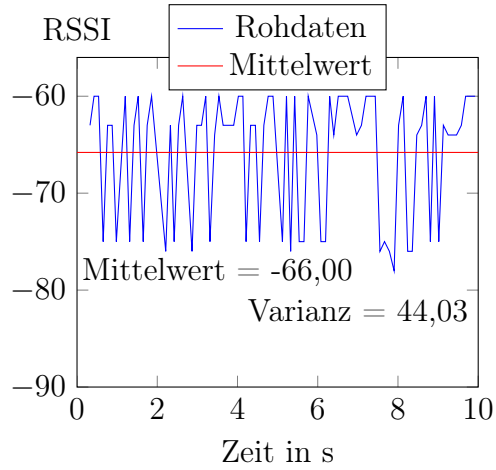
Bei diesen Versuchen zeigt sich, dass die Ausrichtung bei einem Meter Entfernung auf den Mittelwert nur einen geringfügigen Einfluss besitzt. Bereits bei Vergleich des Wertebereichs und der Varianz in den Abbildungen 2.6a und 2.6b ist jedoch zu sehen, dass die Signalqualität durch die veränderte Ausrichtung erheblich beeinträchtigt wird. Der Wertebereich der gemessenen Signale steigt von  $[-64, -68]$  bei einer direkten Ausrichtung auf  $[-60, -78]$ . Die Varianz steigt von 1,47 auf 44,03. Trotz der geringen Abweichungen im Mittelwert lässt sich daher sagen, dass auch bei kurzen Distanzen zwischen Sender und Empfänger die Signalqualität erheblich von der Ausrichtung beeinflusst wird. Ein ähnliches Verhalten ist bei einem Abstand von zwei Metern zu beobachten. Dabei verschlechtern sich jedoch nicht nur Varianz und Wertebereich, sondern auch der Mittelwert. Er sinkt von  $-72,07$  in Abbildung 2.6c auf  $-78,46$  in Abbildung 2.6d.

Ab einer gewissen Entfernung scheinen die ankommenden Signalstärken hauptsächlich von Reflexionen beeinflusst zu werden. Bei einer Entfernung von ca. 9,5 m ist zu sehen, dass die Varianz bei einer um 90 Grad gedrehten Ausrichtung abnimmt, anstatt wie zuvor zu steigen. Ebenso sinkt die Spannweite des Wertebereichs. Der Mittelwert sinkt durch die Drehung von  $-75,6$  auf  $-81,44$ . Der Mittelwert verhält sich daher wie bei einer Entfernung von zwei Metern.

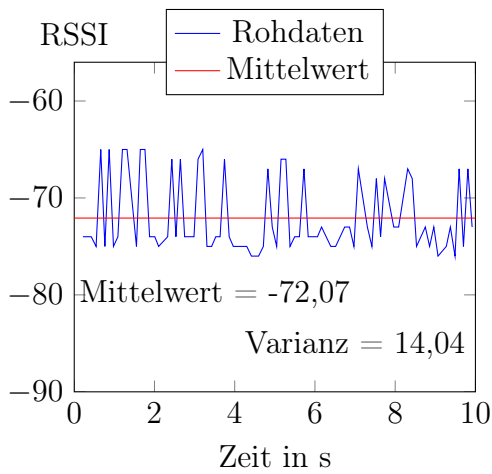
Anhand dieser Graphen ist bereits ein grundlegendes Problem für die Lokalisierung von beweglichen Beacons zu erkennen. Je nach Ausrichtung des Senders variieren die Parameter, an denen ein bestimmter Abstand gemessen werden kann, stark. Darauf ist unter anderem auch die geringe erzielte Genauigkeit anderer wissenschaftlicher Arbeiten



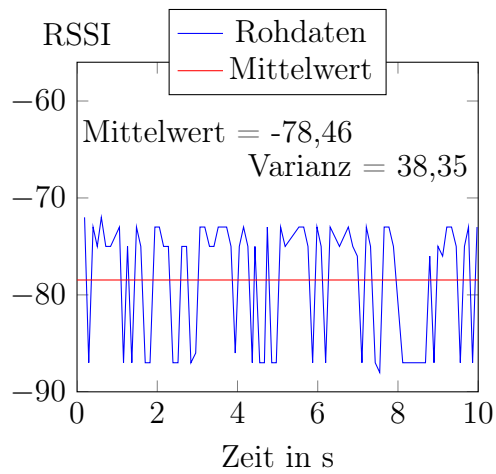
(a) Abstand 1 m



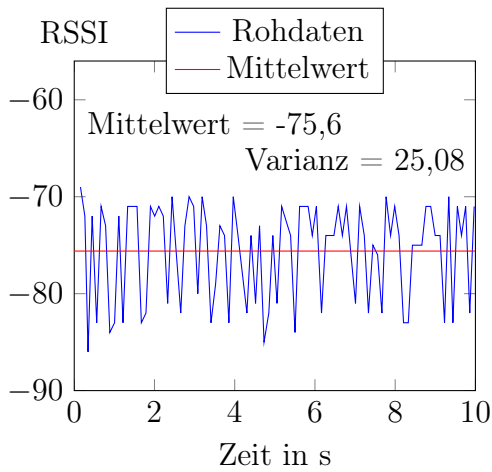
(b) Abstand 1 m



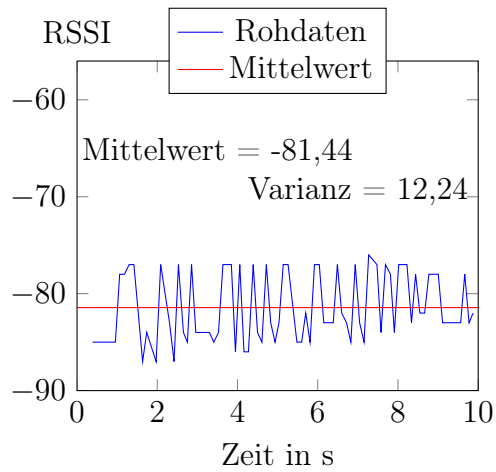
(c) Abstand 2 m



(d) Abstand 2 m



(e) Abstand 9,5 m



(f) Abstand 9,5 m

Abbildung 2.6: Ergebnisse einer Messung mit verschiedenen Abständen zwischen Sender um Empfänger über 10s. In der linken Spalte sind diese in einer geraden Linie aufeinander ausgerichtet, in der rechten Spalte um 90 Grad gedreht.

zurückzuführen. Deshalb kann im Rahmen dieser Arbeit nicht damit gerechnet werden, eine Genauigkeit von zwei bis drei Metern zu unterbieten. Des Weiteren lässt sich anhand der Ergebnisse von [Tre15] sagen, dass bereits geringe Änderungen in der Ausrichtung des Beacons zu Ortungsungenauigkeiten führen können.

Ein bisher noch nicht erwähnter Einfluss auf die Bluetooth-Signalqualität ist das WLAN-Signal. Fahrager et al. [FH14] konnten in ihrer Ausarbeitung zeigen, dass gemessene Bluetooth-Signale auf einem Smartphone durch eingeschaltetes WLAN zum Negativen hin beeinflusst werden. Dadurch traten häufig kleine Ausreißer in einer Messreihe auf, die bei abgeschaltetem WLAN nicht mehr vorhanden waren. Dieses Problem lässt sich auf die Raspberry Pis übertragen. Die USB-Ports, in denen die Adapter stecken, sitzen direkt nebeneinander. Somit befinden sich auch Bluetooth-Adapter und WLAN-Antenne direkt nebeneinander. Dadurch kann die WLAN-Kommunikation das eingehende Bluetooth-Signal stören. Solange eine Netzwerkverbindung über WLAN genutzt werden soll, ist dieses Problem nicht zu lösen.

## 2.5 Ortsbestimmung über Fingerprinting

Grundsätzlich können Ortungsverfahren in die Bereiche Fingerprinting- und Trilaterationsverfahren eingeordnet werden [Mey09]. Aufgrund der vielen Faktoren, die auf die Signale einwirken, besitzen Verfahren, die Fingerprinting nutzen, in der wissenschaftlichen Literatur in den meisten Fällen eine höhere Genauigkeit als Trilaterationsverfahren. Da im zu erweiternden System bereits ein Verfahren im Einsatz ist, das auf einem Fingerprinting-Ansatz über Support Vector Machines basiert, wird nicht weiter auf Trilaterationsverfahren eingegangen, sondern ausschließlich Fingerprinting vorgestellt.

### 2.5.1 Allgemeines

Das Prinzip der Fingerprinting-Verfahren lässt sich allgemein auf zwei Phasen reduzieren. In der Offline-Phase werden Vergleichsdaten an verschiedenen Punkten im Raum bzw. Gebäude aufgenommen. Darüber lassen sich Eigenschaften eines an dieser Stelle gemessenen Signals in den Vergleichsdaten abbilden. Diese Eigenschaften – so die Theorie – sind für jeden Ort der Aufnahme einzigartig. So lassen sich (Finger-)Abdrücke für jeden einzelnen Bereich oder sogar für jede Position zum Beacon ermitteln.

In diesen Eigenschaften sind ebenfalls viele der möglicherweise auftretenden Störungen enthalten. Das heißt, solange es sich um für einen Ort signifikante Störungen handelt, wie zum Beispiel durch Wände, tragen sie dazu bei, den Fingerabdruck realistischer zu machen. Schwer bzw. bisher unmöglich in einem Fingerabdruck abzubilden, ist die dynamische Interaktion mit Menschen.

Der zeitaufwändigste Teil des Fingerprintings besteht im Anlegen eines solchen Abdrucks. Gerade zur Positionsbestimmung ist es wichtig, ausreichend genaue Fingerabdrücke anzulegen. Dabei sollte nicht nur auf die Position der einzelnen Messungen geachtet werden, sondern auch darauf, dass die Daten möglichst realitätsgetreu sind. Das bedeutet, dass, wenn später ein Mensch mit einem Beacon lokalisiert werden soll, das



Beacon sich beim Anlegen der Vergleichsdaten an der gleichen Position am Menschen befinden sollte, wie in der späteren Ortung. Des Weiteren sollte darauf geachtet werden, dass zusätzlich die Bewegung einer Person in den Vergleichsdaten abgebildet wird. Je realitätsgetreuer die Fingerabdrücke zu einzelnen Positionen angelegt werden, desto besser kann später das Ergebnis einer Ortung werden.

In der Online-Phase werden statt längeren Abdrücken nur kurze Zeitintervalle aufgenommen. Die so aufgenommenen Daten können danach über verschiedenste Parameter mit den Fingerabdrücken verglichen werden. Je näher die Daten an einem Fingerabdruck liegen, desto wahrscheinlicher ist es, dass die Aufnahme tatsächlich aus diesem Bereich stammt.

Die gängigsten Ansätze zur Zuordnung von Bereichen zu Signal-Captures sind K-Nearest-Neighbour (KNN) und der Bayes-Klassifikator. Beim KNN-Algorithmus wird ein Datum anhand seiner  $k$  umliegenden Nachbarpunkte einer bestimmten Klasse zugeordnet. Dabei können unterschiedliche Abstandsmaße wie der euklidische Abstand oder die Manhattan-Metrik verwendet werden. Ebenfalls besteht die Möglichkeit, einzelne Nachbarn verschieden stark zu gewichten. Neben diesen populären Klassifikatoren können theoretisch beliebige andere Klassifikatoren verwendet werden. Wichtig ist zunächst nur, dass ein Datum einer Klasse von zuvor aufgenommenen Daten zugeordnet wird.

### 2.5.2 Fingerprinting mit Support Vector Machines

Support Vector Machines (SVMs) gehören zu den linearen Binärklassifikatoren im Bereich des *Supervised Machine Learning*. Das bedeutet, dass genau wie im Falle eines Fingerprinting-Verfahrens zunächst ein Vergleichsdatensatz aufgenommen bzw. bereitgestellt werden muss. Ein solcher Datensatz wird Trainingsdatensatz genannt. Für die Trainingsdaten ist bereits vor der Klassifikation bekannt, zu welcher Klasse ein jedes Datum gehört.

Die Rohdaten können jedoch nicht direkt in den Klassifikator gegeben werden. Dazu werden die Rohdaten durch beliebige Verfahren in Merkmals-Vektoren transformiert. Die Gesamtheit aller Merkmals-Vektoren bildet den Merkmalsraum. Daher ist die Erstellung der Merkmals-Vektoren entscheidend für die Genauigkeit der Klassifikation über eine SVM.

Je besser sich die Merkmale der einzelnen Klassen trennen lassen, desto einfacher wird die Klassifizierung der Daten in einer späteren Online-Phase. Anschaulich ist die lineare Trennbarkeit von zwei Klassen im Zweidimensionalen genau dann gegeben, wenn eine Gerade so positioniert werden kann, dass sie beide Klassen exakt trennt.

Im mehrdimensionalen Fall wird die Trennung der Klassen durch eine Hyperebene gewährleistet. Die Lösung ist ohne weitere Bedingungen nicht eindeutig. Daher wird als Lösung die Hyperebene gewählt, zu der alle Punkte aus dem Merkmalsraum den maximalen kleinsten Abstand besitzen (siehe Abbildung 2.7). Dabei ist  $H_3$  die Hyperebene, die die Bedingung erfüllt. Dadurch soll ein Sicherheitsbereich für die Online-Phase generiert werden, der dafür sorgt, dass auch Punkte im Randbereich einer Klasse zuverlässig zugeordnet werden können, auch wenn diese nicht im Trainingsdatensatz enthalten waren.

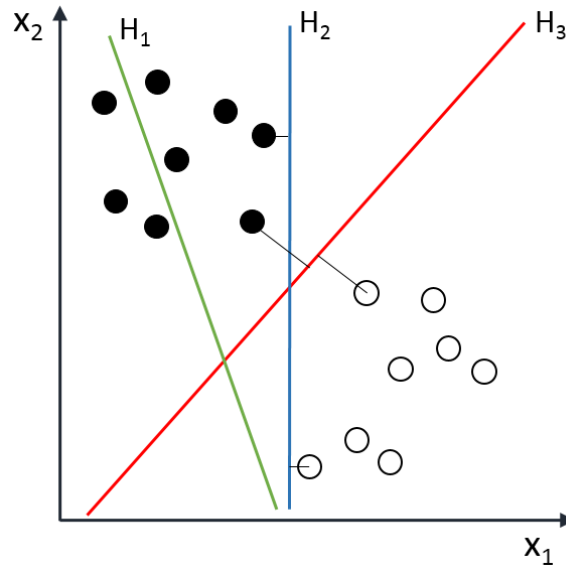


Abbildung 2.7: Verschiedene Möglichkeiten zur Separierung von zwei Klassen.  $H_1$  liefert keine Separierung,  $H_2$  liefert eine mögliche Separierung und  $H_3$  liefert die Separierung mit maximalem kleinsten Abstand aller Punkten.

In der Realität lassen sich allerdings die meisten Klassen bereits im zweidimensionalen nicht perfekt trennen, da häufig Überlappungen z.B. durch fehlerhafte Messdaten vorhanden sind. Über den sogenannten Kernel-Trick lässt sich dieses Problem lösen. Dieser beruht auf der Eigenschaft, dass eine künstliche Erhöhung der Dimension des Merkmalsraums dazu führt, dass ab einer genügend hohen Dimension die Trennbarkeit der Klassen über eine Hyperebene gegeben ist.

Da diese Transformation sehr rechenaufwendig ist und die Rücktransformation zu unbrauchbaren Ergebnissen führen kann, darf der Kernel nicht beliebig gewählt werden, sondern muss symmetrisch und positiv definit sein. Dadurch lässt sich der Rechenaufwand erheblich senken. Die in der Praxis am häufigsten verwendeten Kernel basieren auf Radial Basis Functions (RBF) und deren Verbesserung, den PUK-Kernen.

Diese erzielen in der Praxis mindestens gleich gute Klassifikationen, in der Regel jedoch bessere [AY14]. In beiden Fällen ist der Grundgedanke, dass die Transformation über eine Glockenkurve erfolgt. Bei PUK-Kernen lässt sich diese über mehrere Parameter anpassen, während RBF-Kernel nur einen Parameter zur Verfügung haben.

Wird ein beliebiger Klassifikator allerdings so genau an die Trainingsdaten angepasst, dass sich beide Klassen garantiert vollständig trennen lassen, führt dies in einer realen Anwendung dazu, dass während der Online-Phase Daten häufig schlechter klassifiziert werden. Es findet eine Überanpassung des Klassifikators an die Trainingsdaten statt. Dies kann durch die Einführung zusätzlicher Parameter bzw. der richtigen Parameterwahl gelöst werden. Dabei wird zugelassen, dass im Training einzelne Daten falsch einsortiert werden. Somit wird eine größere Robustheit gegenüber Messfehlern erreicht.

Des Weiteren können SVMs nur binär klassifizieren. Mit einer einfachen SVM könnte

daher nur entschieden werden, ob sich eine Person in einem Gebäude bzw. Bereich aufhält oder nicht. Erwünscht ist jedoch, dass erkannt wird, in welchem bestimmten Bereich sie sich befindet. Daher muss die SVM in einen One-Vs-All-Klassifikator eingebettet werden, der für jede einzelne Klasse testet, ob ein Datum dieser zuzuordnen ist. Damit lassen sich SVMs ebenfalls auf Probleme mit mehreren Klassen anwenden. Dadurch entsteht die Möglichkeit, eine Support Vector Machine auf das hier behandelte Lokalisierungsproblem anzuwenden.



## 3 Softwaretechnische Umsetzung

In diesem Kapitel wird die Integration der Lokalisierung von beweglichen Beacons in das bisherige System der Fachhochschule Aachen konzipiert und realisiert. Dazu wird zunächst das bisherige Anwendungsszenario und dessen softwaretechnische Umsetzung dargestellt. Danach werden die Anpassungen erläutert, die zur Integration der Beacon-Ortung notwendig waren. Ebenso wird erläutert, wie die Synchronisierung der Empfänger und der im Server ankommenden Datenpakete abläuft.

### 3.1 Aufbau des Testsystems

Die Fachhochschule Aachen verfügt bereits über ein Testsystem zur Ortung von beweglichen Empfängern mit stationären Sendern, dessen Aufbau und Funktionsweise im Folgenden verdeutlicht werden soll, bevor dieses um die Ortung beweglicher Sender mit stationären Empfängern erweitert wird. Das Testsystem besteht im Wesentlichen aus drei Paketen, die zur Ortung beweglicher Sender angepasst werden müssen, sowie einer Android-App, die zum Lokalisieren und Navigieren eines Smartphones oder Tablets als beweglicher Empfänger genutzt wird.

Auf die App wird in dieser Ausarbeitung nicht weiter eingegangen werden, da sie keinen Einfluss auf die Ortung beweglicher Sender besitzt, sondern – stark vereinfacht dargestellt – lediglich die empfangenen Datenpakete an einen Server mit REST-API weiterleitet und dessen Antworten auf dem Smartphone darstellt.

Die drei für die Positionsbestimmung mit beweglichen Sendern und stationären Empfängern wichtigen Pakete sind der *Location-Solver*, die *Data-Objects* und die *Backend-Webservices*, die nach einer Erklärung des grundsätzlichen Ablaufs detaillierter betrachtet werden sollen.

#### 3.1.1 Ortungsablauf mit beweglichen Empfängern und stationären Sendern

Der grundsätzliche Ablauf zur Positionsbestimmung ist in verschiedene Phasen einzuteilen. Noch vor der Lokalisierung muss ein Gebäude im OpenStreetMap-Format kartiert werden. Das heißt, es müssen möglichst genaue Gebäudeinformationen zur Verfügung gestellt werden, um die Position korrekt anzeigen zu können.

Nach der Kartierung müssen einzelne Räume oder größere Abschnitte in kleinere Bereiche unterteilt werden. Diese besitzen zum Zeitpunkt dieser Ausarbeitung in der Regel eine Größe von ca.  $3 \times 3$  Metern. Danach sind die Kartendaten vollständig und liegen auf dem Server bereit. Sie werden hauptsächlich zur Visualisierung des Standortes benötigt.

Sollte ein Smartphone in einer Live-Ortung einem Gebiet zugeordnet werden, wird das Zentrum des Gebietes mit einer blauen Markierung versehen.

Bevor eine solche jedoch stattfinden kann, müssen die Beacons installiert werden. Um eine Positionsbestimmung zu gewährleisten, muss sich das Smartphone zu jedem Zeitpunkt in der Sendereichweite von mindestens drei Beacons befinden. In den Versuchsaufbauten werden in der Regel vier Beacons pro Raum verwendet, da die Wände im Versuchsflügel der Fachhochschule (01c) sehr dick sind.

Dadurch wäre theoretisch auch eine Ortung im dreidimensionalen möglich, die jedoch nicht durchgeführt wird. Nach der Installation der Beacons folgt die Offline-Phase des Fingerprintings. Dabei müssen für jeden  $3 \times 3$  Meter-Bereich möglichst über das gleiche Zeitintervall – meist über 30 Sekunden bis zu einer Minute – Referenzdaten über das Smartphone aufgenommen und an einen Server gesendet werden.

Mit diesen Daten kann im Server eine Support Vector Machine als Fingerprinting-System trainiert werden. Sobald der Empfänger lokalisiert werden soll, wird ein Datensatz über ca. eine Sekunde aufgenommen und ebenfalls an den Server geschickt. Durch die bereits trainierte SVM kann der Datensatz klassifiziert werden. Dies bedeutet, dass er genau einem Bereich zugeordnet wird. Die Lokalisierung war erfolgreich, wenn die Person tatsächlich in diesem Bereich steht.

Dabei ist das Ziel, dass möglichst keine Fehlortung auftritt, was allerdings nicht immer garantiert werden kann. Wurde eine Klassifizierung des Datensatzes durchgeführt, sendet der Server die Antwort an das Smartphone zurück. Dort werden die Ergebnisse auf einer Karte angezeigt. Zusätzlich besitzt der Server ebenfalls die Möglichkeit die Ortungsergebnisse über eine Webanwendung auf einer Karte darzustellen.

Dies wird insbesondere in dem hier untersuchten Szenario mit beweglichen Beacons benötigt, da Beacons selbst keine Visualisierungsmöglichkeit besitzen. In der App lässt sich neben der einfachen Lokalisierung ebenfalls eine Navigation zu verschiedenen Punkten innerhalb des Gebäudes durchführen. Dies ist für Beacons während dieser Ausarbeitung nicht vorgesehen.

#### 3.1.2 Data-Objects

Bevor detaillierter auf die Funktionsweise der Lokalisierung eingegangen werden kann, müssen die Datenobjekte näher betrachtet werden, insbesondere die, die zur Lokalisierung benötigt werden. Das wichtigste Datenobjekt zur Lokalisierung ist im Testsystem die Klasse *Signal-Capture*, deren UML-Diagramm sich in Abbildung 3.1 findet.

Sie besteht lediglich aus einer Multimap von Mac-Adressen und zugehörigen RSSI-Werten. Fachlich betrachtet, beinhaltet eine Signal-Capture alle in einem bestimmten Zeitintervall (*Capture-Intervall*) empfangenen RSSI-Werte. Diese werden dem Beacon als Absender über die Mac-Adresse zugeordnet. Als Schlüssel in der Multimap werden aus diesem Grund die Mac-Adressen der Beacons gewählt.

Darüber lassen sich die zugehörigen, empfangenen RSSI-Werte in der Klasse *Signal-Capture* verwalten. Neben der einfachen Verwaltung der Daten, ermöglicht die Klasse mit einer Reihe weiterer Funktionen statistische Eigenschaften, wie die Anzahl der

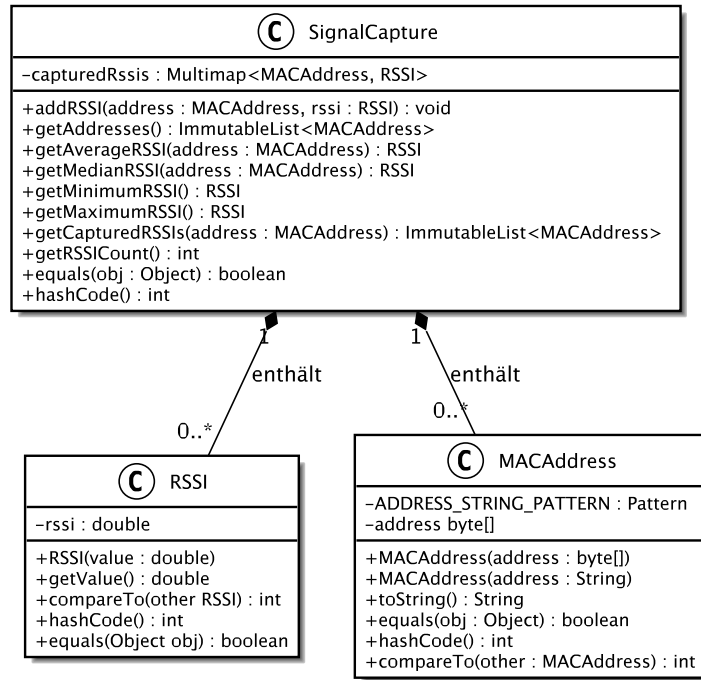


Abbildung 3.1: UML-Diagramm der wichtigsten Datenstrukturen in Data-Objects

empfangenen Signale eines Beacons, arithmetisches Mittel, Minimum oder Maximum auszuwerten.

Die dazu benötigten Hilfsklassen *RSSI* und *MACAddress* werden lediglich zu Verwaltungszwecken benötigt. Sie stellen vor allem sicher, dass die korrekten Formate und Wertebereiche eingehalten werden. Außerdem lassen sich alle genannten Klassen über das *Jackson 2*-Framework in das JSON-Format serialisieren. Mit Hilfe der Serialisierbarkeit lassen sich Objekte, beispielsweise zwischen App und Server, austauschen.

Auf alle weiteren Klassen wird nicht eingegangen, da sie ausschließlich zur Kommunikation über die REST-API des Backends genutzt werden und somit für die Lokalisierung nicht relevant sind.

### 3.1.3 Location-Solver

Der Location-Solver bildet das Herz der eigentlichen Ortung. In diesem Paket findet die Klassifizierung der einzelnen Signal-Captures während der Live-Lokalisierung statt. Aktuell enthält das Paket nur einen Lokalisierer, der die Klassifizierung der Signal-Captures mithilfe des Classifier-Interfaces der JSAT-Bibliothek durchführt. Auf eine eigene Implementierung einer Support Vector Machine kann daher verzichtet werden.

Im Fokus dieses Pakets steht die Umwandlung der Signal-Captures in einen geeigneten Merkmalraum bzw. Merkmal-Vektoren. Nach außen ist die Umwandlung fast vollständig gekapselt, wie in Abbildung 3.2 zu sehen ist. Der *SimpleLocalizer* lässt sich einzig über ein Builder-Pattern erstellen.

Dabei müssen angegeben werden, die Art des Klassifizierers, welche Trainingsdaten

und welche Transformatoren benutzt werden sollen. Die Transformationen werden jeweils auf eine einzelne Signal-Capture angewandt. Dadurch entsteht pro Transformator und Beacon ein Eintrag in einer Zeile eines Merkmal-Vektors. Die Anzahl der Zeilen ist gleich der Anzahl Signal-Captures, die in den Trainingsdaten pro Beacon enthalten ist.

Des Weiteren existiert die Möglichkeit, ein gleitendes Fenster über mehrere Signal-Captures zu definieren, sodass mehrere aufeinander folgende Signal-Captures für die Transformation anstelle einer einzelnen Signal Capture genutzt werden. Innerhalb des *SimpleLocalizer* können die Signal-Captures und deren RSSI-Werte über verschiedene Implementationen des *RssiTransformers* in den Merkmalraum transformiert werden.

Im Zuge dieser Ausarbeitung werden weitere Möglichkeiten (siehe Kapitel 4) hinzugefügt, Signal-Captures in einen Merkmalraum als Merkmal-Vektoren zu transformieren, da bisher nur der Mittelwert als Transformator zur Verfügung steht.

Die wichtigsten beiden Methoden des *SimpleLocalizers* sind *train* und *locate*. Innerhalb der *train*-Methode wird der Klassifikator trainiert. Das bedeutet, dass aus den zuvor aufgenommen Daten aus den verschiedenen Raumbereichen ein Fingerabdruck erstellt wird. Der *SimpleLocalizer* hat die dazu notwendigen Daten über den zugehörigen Builder bereits während seiner Erstellung erhalten, sodass sie lediglich über die gewünschten Transformatoren in einen Merkmalraum überführt werden müssen.

Wichtig dabei ist, dass für alle Trainingsdaten bekannt sein muss, aus welchen Raumbereichen sie stammen. Die Anzahl der Raumbereiche ist gleich der zu unterscheidenden Klassen. In der *locate*-Methode spielt sich die Online-Phase des Fingerprintings ab. Soll ein entsprechendes Gerät lokalisiert werden, wird eine *deviceId* und eine Signal-Capture aus dem letzten Zeitintervall benötigt. Die *deviceId* kann beliebig gewählt werden. Häufig wird zur besseren Verständlichkeit die String-Repräsentation der Mac-Adresse genutzt.

Die Signal-Capture muss zur Klassifizierung auf die exakt gleiche Art und Weise wie zuvor die Trainingsdaten in den Merkmalraum transformiert werden, damit sich alle Daten im gleichen mathematischen Raum befinden. Danach wird die Klassifikation über einen One-Vs-All-Klassifikator mit der SVM durchgeführt.

Die daraus entstehenden Resultate werden in einer Liste von *LocalizationResults* gespeichert. Sie besteht aus einem Resultat pro Klasse. Darin enthalten sind der Name des Raumbereichs und die zugehörige Konfidenz, die ein Maß angibt, wie sicher der Klassifikator ist, dass die Signal-Capture in diesen Bereich bzw. zu dieser Klasse gehört.

#### 3.1.4 Backend-Webservices

Über die *Backend-Webservices* wird die Steuerung der Lokalisierung und die Interaktion mit Benutzern über eine Rest-API ermöglicht. Ebenso kann mit dem Backend über eine Webanwendung (vgl. Abbildung 3.3) interagiert werden. Bisher wird diese jedoch ausschließlich zu Test- und Verwaltungszwecken verwendet. Außerdem bietet die Webanwendung eine Visualisierung der ermittelten Position auf einer Open Street Map Karte. Zur Steuerung der Abläufe über Rest-Aufrufe werden intern verschiedene Steuerungseinheiten (Controller) des Spring Boot Frameworks genutzt. Die Zuordnung der Aufrufe findet auf der Softwareseite über die entsprechenden Spring-Annotationen statt.



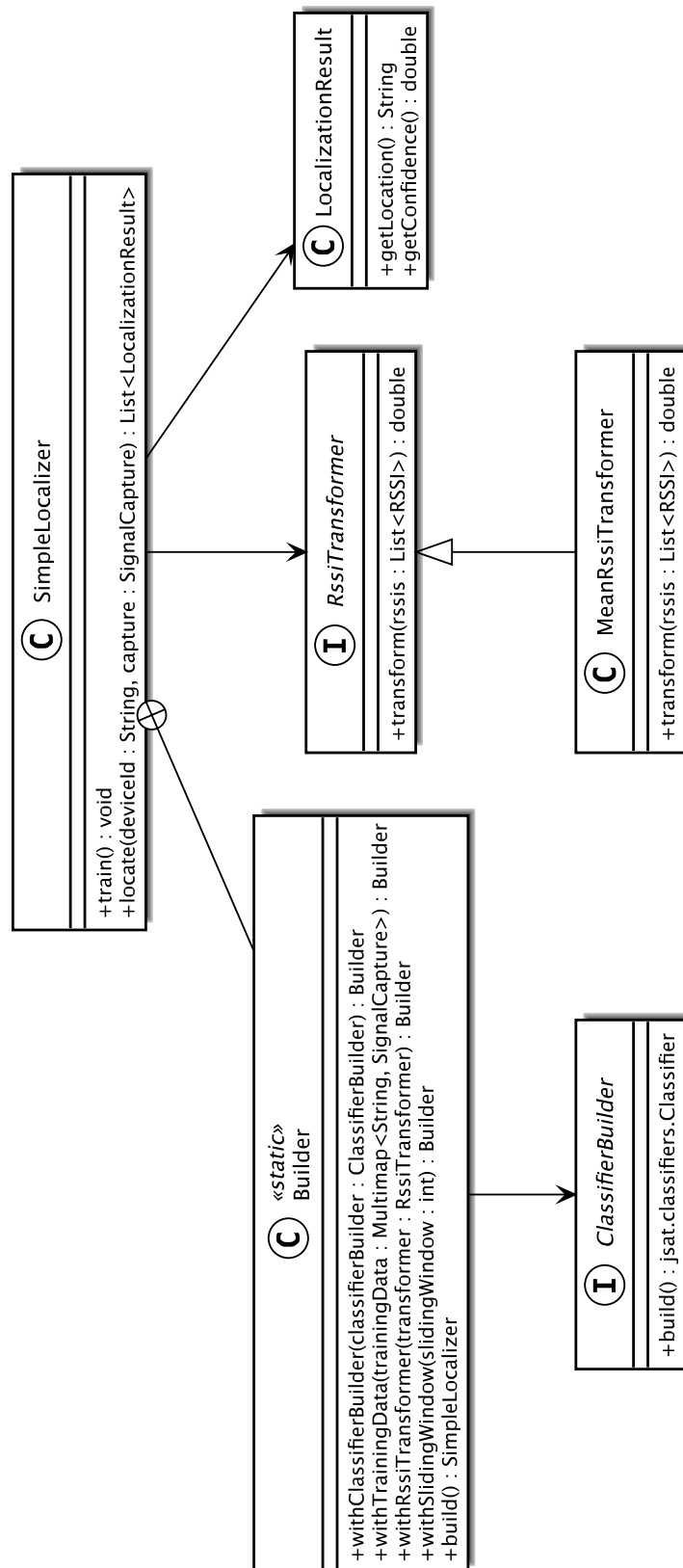


Abbildung 3.2: UML-Diagramm der Datenstrukturen im Location-Solver

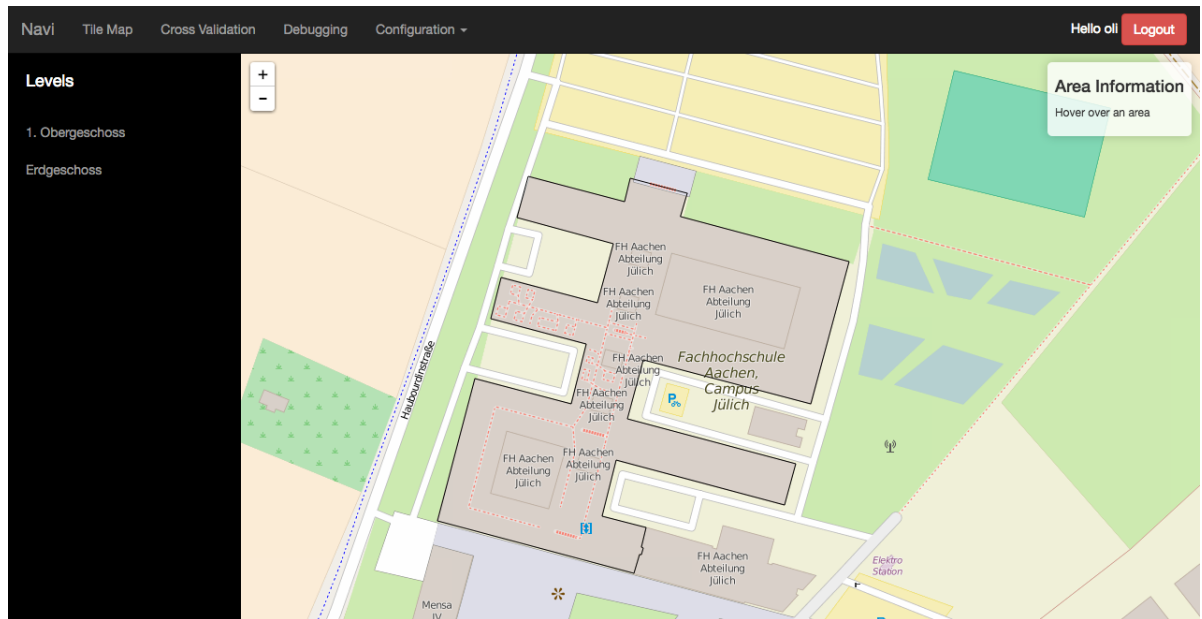


Abbildung 3.3: Grafische Oberfläche der Webanwendung. Abgebildet ist die Visualisierung der Karte im Reiter „Tile Map“

Auch in diesem Abschnitt wird nur auf die zur Lokalisierung benötigten Rest-Controller eingegangen. Sie bilden im Wesentlichen die Schnittstelle zwischen der Rest-API und dem *Location-Solver*. Die dafür zuständigen Controller sind der *LocalizationRestApiController* und der *SignalCaptureRestApiController*, die die verschiedenen Phasen des Fingerprinting widerspiegeln.

An den *SignalCaptureRestApiController* werden die gesammelten Daten der Offline-Phase über den entsprechenden Rest-Aufruf weitergeleitet. Danach werden die Daten einer Plausibilitätsüberprüfung unterzogen, um auszuschließen, dass zum Beispiel nicht kartierte Raumbereiche angegeben wurden. Danach werden die Daten auf dem Server in Datenzugriffsobjekten (engl. Data Access Objects, DAO) im JSON-Format abgespeichert und für Lokalisierungen vorgehalten.

Der *LocalizationRestApiController* ist für alle Anfragen, die die Online-Phase betreffen, zuständig und bildet die Schnittstelle zwischen *Location-Solver* und den Nutzern der Lokalisierung. Dazu werden die benötigten Methoden *train* und *locate* des *Simple-Localizers* mit Hilfe von Annotationen über Rest-Aufrufe aufrufbar gemacht.

Zur Durchführung der Online-Phase, bei der in Echtzeit aufgenommene Signal-Captures an die *locate*-Methode übergeben werden, muss der Lokalisierer zuvor mit den Daten, die im Server durch den *SignalCaptureRestApiController* gespeichert wurden, trainiert werden. Danach können in Echtzeit Signale klassifiziert und der ermittelte Bereich auf der Karte durch eine blaue Markierung (vgl. Abbildung 4.1) angezeigt werden.

## 3.2 Integration der Beacon-Ortung

Um die Ortung von beweglichen Beacons in das vorhandene System zu integrieren, muss zunächst gelöst werden, wie die Kommunikation der Empfänger synchronisiert werden kann, um Aufträge zu erhalten bzw. aufgenommene Daten in der Online-Phase synchron zu versenden. Ebenso müssen die empfangenen Signale auf dem Server synchronisiert und verarbeitet werden. Die Lösungsansätze und deren Umsetzung werden in diesem Abschnitt erläutert. Außerdem wird ein erster Funktionstest durchgeführt.

### 3.2.1 Synchronisation der Empfänger

Bevor Beacons geortet werden, können muss noch ein grundlegendes Problem gelöst werden, das im klassischen Szenario, wo Empfänger zu orten sind, nicht auftritt. Dabei handelt es sich um die Synchronisation aller zur Verfügung stehenden Empfänger. Zur Verfolgung von beweglichen Sendern mit stationären Empfängern müssen alle Empfänger sowohl gleichzeitig angesteuert werden können, als auch gleichzeitig senden.

Daraus resultieren zwei Anforderungen: Erstens müssen alle Raspberry Pis gleichzeitig Befehle empfangen und die eingegangenen Bluetooth-Signale versenden können, ohne dass ein Rückstau oder ein Datenverlust auftritt. Zweitens müssen diese Daten vom Backend verarbeitet werden können. Außerdem muss die Netzwerkkapazität ausreichend sein, sodass alle Daten mit nur minimaler Latenz übertragen werden.

In einer vorausgehenden Untersuchung zu diesem Problem wurde ermittelt, dass der Einsatz eines RabbitMQ-Servers diese Probleme weitgehend löst. Bei RabbitMQ handelt es sich um einen Message-Broker, der das frei verfügbare AMQP-Protokoll einsetzt. Dadurch existieren in den meisten Programmiersprachen Bibliotheken, die die Kommunikation mit einem solchen Server ermöglichen. Daher sind auch die benötigten Java- und NodeJS-Anbindungen als freie Software verfügbar.

Die Kommunikation über AMQP folgt einen *Publisher/Subscriber*-Muster, ähnlich dem von Newslettern. Für einen Newsletter kann sich ein Benutzer registrieren und immer wenn ein neuer Newsletter erscheint, wird dieser an jeden registrierten Benutzer geschickt. Ähnlich läuft das Verschicken von Nachrichten über AMQP ab. Ein *Subscriber* registriert sich an einem *Exchange* als Empfänger der Nachrichten und ein *Publisher* als Sender.

Jedes mal, wenn eine neue Nachricht vom Publisher an den Exchange gesendet wird, erhalten alle Subscriber diese Nachricht. Dabei können sie verschiedene weitere Optionen nutzen, das grobe Prinzip bleibt allerdings erhalten. Nachrichten, die durch einen Exchange geleitet werden, landen in Queues, die nach dem FIFO-Prinzip abgearbeitet werden.

Dadurch können alle Raspberry Pis nahezu gleichzeitig (Netzwerklatenz) angesteuert werden. Außerdem können so alle Raspberry Pis gleichzeitig ihre gemessenen Werte an den Server schicken, ohne dass Nachrichten verloren gehen, da sie in einer Warteschlange landen, bis sie an das Backend weitergereicht werden. Auch dies geschieht in wenigen Millisekunden. Da zwischen einzelnen Nachrichten der Raspberry Pis mindestens 500 ms liegen müssen, um sinnvolle Messergebnisse zu erzielen, kann diese Latenz vernachlässigt

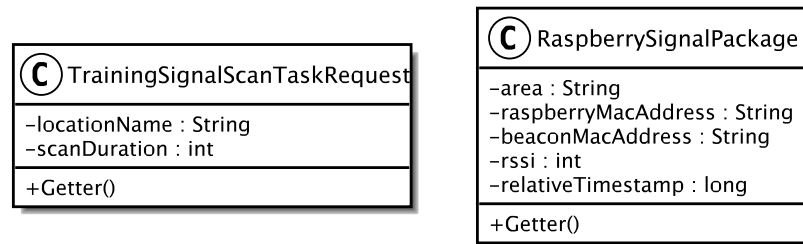


Abbildung 3.4: UML-Diagramm der neu hinzugefügten Datenstrukturen zur Kommunikation

werden, solange die Nachrichten im Backend ausreichend schnell abgearbeitet werden können.

Beim Ermitteln der Trainingsdaten wird ein anderer Exchange genutzt als bei der Online-Phase. Zusätzlich muss zum Starten einer Datenerhebung für Trainingsdaten bekannt sein, wie lange eingehende Signale aufgenommen werden sollen und aus welchem Bereich die Daten stammen. Dazu müssen die Data-Objekte um eine entsprechende Datenstruktur erweitert werden. Dies wird in der neuen Klasse *TrainingSignalScanTaskRequest* umgesetzt.

Sie enthält, wie in Abbildung 3.4 zu sehen ist, einen String für den Bereich und einen Integer-Wert für die Aufnahmedauer in Sekunden. Sie ist im JSON-Format serialisierbar und lässt sich so über den RabbitMQ-Server versenden. Auf dem Raspberry werden die entsprechenden Daten ausgewertet und als Scan für die gewünschte Dauer umgesetzt.

Während die Messung läuft, wird über die in Abschnitt 2.3 beschriebene Noble-Schnittstelle die Daten erhoben. Bei jedem eingehenden Bluetooth-Signal werden jedoch, außer der Mac-Adresse des Beacons und des gemessenen RSSI-Wertes, ebenfalls das angegebene Gebiet, die Mac-Adresse des Raspberry Pis und ein relativer Zeitstempel – in den Raspberry Pis wird eine interne Uhr gestartet, sobald Trainingsdaten erhoben werden – in Millisekunden in einer Liste gespeichert. Ist die Zeit für die Datenerhebung abgelaufen, wird die Liste mit allen Daten im JSON-Format serialisiert und über einen weiteren Exchange an den Server geschickt, wo diese weiterverarbeitet werden. Auf die genaue Bedeutung der Werte im *RaspberrySignalPackage* wird im nächsten Abschnitt eingegangen.

Der Start der Online-Phase funktioniert ebenso über einen eigenen Exchange mit dem Unterschied, dass keine Datenstruktur benötigt wird. Sobald eine beliebige Nachricht über diesen Exchange eingeht, wird ein Bluetooth-Scan gestartet, der solange läuft bis er abgebrochen wird. Dadurch lassen sich alle Nachrichten unabhängig von Sender oder Empfänger und ohne Laufzeiteinbußen synchronisieren.

Listing 3.1: Über Spring injizierte RabbitMQ-Abhängigkeiten

```

1  @Autowired
2  private RabbitTemplate template;
3
4  @Qualifier(value = "trainingTaskFanout")
5  @Autowired
6  private FanoutExchange trainingFanoutExchange;
7
8  @Qualifier(value = "localizeRequestFanout")
9  @Autowired
10 private FanoutExchange localizationFanoutExchange;

```

Listing 3.2: Ein Template für Nachrichten über RabbitMQ. In der Konfiguration wird der Verbindungstyp und die Konvertierer übergeben.

```

1  @Bean
2  public RabbitTemplate rabbitTemplate() {
3      RabbitTemplate rabbitTemplate = new RabbitTemplate(connectionFactory());
4      rabbitTemplate.setMessageConverter(jsonMessageConverter());
5      return rabbitTemplate;
6  }

```

### 3.2.2 Anpassungen im Backend

Damit die Raspberry Pis aus der Webanwendung gesteuert werden können, muss zunächst die Benutzungsoberfläche um einen entsprechenden Reiter erweitert werden, über den sich die Raspberry Pis steuern lassen. Ebenso muss zu diesem Zweck ein weiterer Controller geschrieben werden, der die Aufgabenübermittlung über eine Rest-Schnittstelle ermöglicht.

Diese Aufgabe wird vom *RaspberrySignalCapturingController* übernommen. Dazu müssen zuerst die entsprechenden *Exchanges* in RabbitMQ angelegt werden. Da die Webanwendung auf dem Spring Framework basiert, kann dazu Spring AMQP verwendet werden, da RabbitMQ dieses Nachrichtenprotokoll unterstützt. Zu diesem Zweck werden über das Spring Framework zuvor konfigurierte *Exchanges* und eine Vorlage (Template, vgl. Listing 3.2) für Nachrichten über RabbitMQ wie in Listing 3.1 injiziert.

Dadurch besteht die reine Kommunikation zum Senden der Befehle lediglich aus wenigen Zeilen Quellcode, wie in Listing 3.3 zu sehen ist. Neben diesen Methoden für die Kommunikation besteht der neu hinzugefügte Controller nur aus einem Integritätstest zu den Formulardaten in Abbildung 3.5. Über die Schaltflächen „Capture Signal“ und „Start Localization“ werden die entsprechenden Befehle an die Raspberry Pis abgesetzt. Bevor die Lokalisierung gestartet werden kann, müssen jedoch wie zuvor Referenzdaten als Fingerabdruck gesammelt und die Support Vector Machine trainiert werden.

Nachdem eine der beiden Methoden aus Listing 3.3 aufgerufen wurde, beginnt das Raspberry mit der gewünschten Aktion. Zunächst soll auf die Offline-Phase eingegangen werden. Dazu wird die Methode *publishTrainingRequestToRabbit* aufgerufen. Dabei werden dem Raspberry Pi die gewünschte Dauer der Aufnahme und das Gebiet mitgeteilt.

Listing 3.3: Publizieren von Nachrichten mit Hilfe eines Templates

```
1 public void publishTrainingRequestToRabbit(TrainingSignalScanTaskRequest ←
   signalScanTaskRequest) throws IOException {
2     template.convertAndSend(trainingFanoutExchange.getName(), "", ←
       signalScanTaskRequest);
3 }
4
5 public void publishLocalizationRequestToRabbit() throws IOException {
6     template.convertAndSend(localizationFanoutExchange.getName(), "", "");
7 }
```

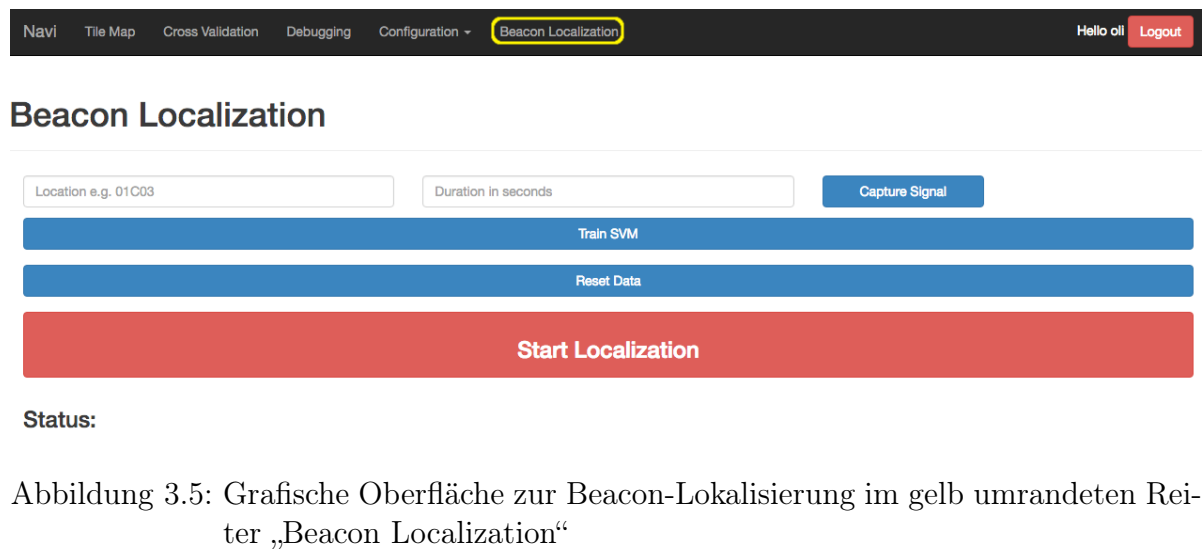


Abbildung 3.5: Grafische Oberfläche zur Beacon-Lokalisierung im gelb umrandeten Reiter „Beacon Localization“

Listing 3.4: Empfangen von Nachrichten mit Fingerabdrücken durch Spring-Annotationen im *SignalCaptureRestController*

```
1 @RabbitListener(queues = "#{trainingReceiverQueue.name}")
2 public void onReceivedRaspberryTrainingSignal(RaspberrySignalPackage[] msg) throws ←
   DAOException {
3     ...
4 }
```

Listing 3.5: Synchronisierung der Trainingsdaten über relative Zeitstempel

```

1 for(String piMacAddress : areaSignalCaptures.keySet()) {
2     int start = intervalStart;
3     int end = intervalEnd;
4     areaSignalCaptures.get(piMacAddress).stream()
5         .filter(raspberrySignalPackage ->
6             raspberrySignalPackage.getRelativetTimestamp() > start)
7         .filter(raspberrySignalPackage ->
8             raspberrySignalPackage.getRelativetTimestamp() <= end)
9         .forEach(raspberrySignalPackage -> capture.addRSSI(
10             new MACAddress(raspberrySignalPackage.getRaspberryMacAddress()),
11             new RSSI(raspberrySignalPackage.getRssi())));
12 }
13 }

```

Die Antwort wird nach Ablauf der gewünschten Zeit über einen weiteren *Exchange* an RabbitMQ gesendet. Im Backend wurde zuvor ein passender Empfänger registriert, der reagiert, sobald Daten über den *Exchange* gesendet werden. Dafür wurde der *Signal-CaptureRestApiController* um die Methode aus Listing 3.4 erweitert. Durch die *Rabbit-Listener*-Annotation wird gewährleistet, dass sie jedes Mal aufgerufen wird, wenn eine Nachricht über den angegebenen Exchange eingeht.

Dies hat jedoch zur Folge, dass die Nachrichten jedes Raspberry Pis einzeln ankommen und somit synchronisiert werden müssen. Des Weiteren besteht die Notwendigkeit, dass die Nachrichten in Signal-Captures konvertiert werden, damit sie zur Lokalisierung verwendet werden können.

Damit die Synchronisation stattfinden kann, werden alle eingehenden Daten solange gespeichert, bis die Daten aus jedem Raspberry empfangen wurden. Danach können sie durch den *RaspberryCaptureSynchronizationService* synchronisiert werden. Dazu werden die aufgenommenen Daten pro Raspberry Pi betrachtet. Da die einzelnen Pakete einen relativen Zeitstempel von Beginn bis Ende der Messung beinhalten, lassen sich alle Pakete aus einem Capture-Intervall, wie in Listing 3.5, in eine Signal-Capture zusammenfassen. Anstelle der Mac-Adresse des Beacons wird in der Anwendung mit stationären Beacons die Mac-Adresse des Raspberrys genutzt. Dadurch lassen sich die bisherigen Datenstrukturen ohne weitere Anpassung verwenden.

Bei einer Live-Lokalisierung wird ähnlich verfahren. Allerdings entfällt die Synchronisation der Daten über die relativen Zeitstempel. Zur Synchronisation ist es ausreichend, dass gewartet wird, bis vom jedem Raspberry die nötigen Daten übertragen wurden. Dies gilt unter der Annahme, dass die Übertragung und Verarbeitung der Daten nur wenige Millisekunden benötigt, da ein Capture-Intervall mindestens 500 ms dauert. Dazu wurde der *LocalizationRestApiController* analog zum *SignalCaptureRestApiController* mit einer entsprechenden Methode erweitert. Die Umwandlung der Daten in Signal-Captures erfolgt ebenso analog.

Zur Lokalisierung wird allerdings eine *deviceId* benötigt. Dazu wird bei beweglichen Beacons die jeweilige Mac-Adresse verwendet. Erst wenn zu jedem Beacon, das geortet werden soll, eine Signal-Capture von jedem Raspberry Pi aufgenommen wurde, kann es lokalisiert werden. Im Rahmen dieser Ausarbeitung wird daher vorausgesetzt, dass jedes

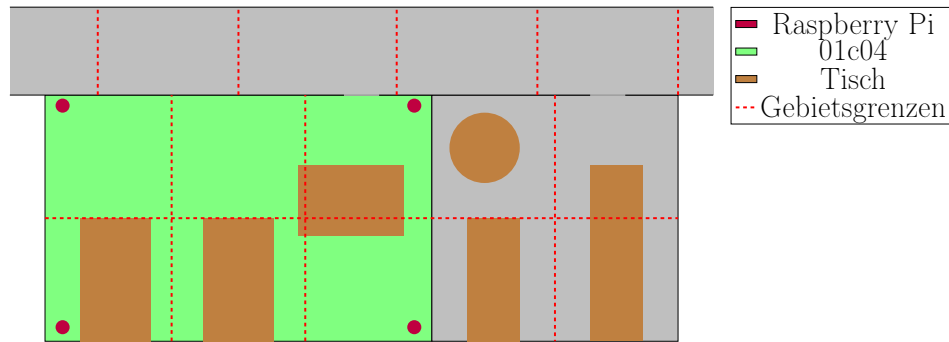


Abbildung 3.6: Die Abbildung zeigt einen nicht maßstabsgetreuen Ausschnitt aus dem Flügel 01c, in dem alle Versuche stattfanden. Die ausgegrauten Bereiche werden für die Lokalisierung beweglicher Beacons nicht eingesetzt, da diese Versuche ausschließlich im grün hinterlegten Raum 01c04 stattfinden. Durch die roten gestrichelten Linien werden die einzelnen Bereiche innerhalb der Räume abgetrennt. Die Nummerierung der Bereiche erfolgt zeilenweise pro Raum von links nach rechts, beginnend mit 01c04\_01 in der oberen linken Ecke und 01c04\_04 in der unteren linken Ecke.

Raspberry zu jedem Zeitpunkt alle Bluetooth-Signale der Beacons empfangen kann.

#### 3.2.3 Anwendungstest

Zur Überprüfung, ob die Integration der Beacon-Lokalisierung in die bestehende Softwareumgebung erfolgreich war, wurde ein einfacher Feldversuch durchgeführt. Dabei wurden die Raspberry Pis jeweils in den Ecken des Raumes 01c04 wie in Abbildung 3.6 positioniert. Danach wurde ein Trainingsdatensatz erstellt, indem zu allen sechs Bereichen im Raum über 30 Sekunden Messdaten erfasst wurden.

Dabei ist die Realitätstreue der erhobenen Daten für das Trainieren der SVM wichtig, um später gute Resultate zu erhalten. Aus diesem Grund wurde das Beacon beim Einmessen des Raumes auf Kopfhöhe platziert, während der Einmessende sich zufällig durch das entsprechende Gebiet bewegte. Diese Messdaten müssen gleichzeitig von allem Empfängern aufgenommen werden, damit der relative Zeitstempel eine Synchronisierung der Daten auf dem Server ermöglicht.

Das Capture-Intervall wurde im ersten Funktionstest auf 500 ms festgelegt. Das bedeutet, dass alle RSSI-Werte, die in dieser Zeit aufgenommen wurden, zu der gleichen Signal-Capture gehören. Dadurch entstehen beim Einmessen 60 Signal-Captures auf jedem Raspberry, die im Fingerprinting als Referenz- bzw. Trainingsdatensatz dienen. Bei den sechs Gebieten enthält der Trainingsdatensatz 360 empfangene Signal-Captures pro Raspberry, die jeweils null bis fünf Messwerte zusammenfassen, da ein Beacon mit einer Frequenz von 10 Hz sendet.

Die Transformation in den Merkmalraum wird über das arithmetische Mittel der Signal-Captures durchgeführt. Ein ähnliches Vorgehen, einen Zeitintervall von 500 ms bis 1000 ms zu wählen und den Mittelwert der RSSI-Werte als Fingerprinting-Vergleichswert



zu benutzen, findet auch bei anderen Autoren Anwendung [SHRB11],[KMU<sup>+</sup>14].

Jedes Raspberry, das als Empfänger benutzt wird, führt zu mindestens einer weiteren Merkmal-Vektor-Spalte. Dies hängt davon ab, wie viele Transformatoren genutzt werden, um die Merkmal-Vektoren aus einer einzelnen Signal-Capture zu generieren. Im ersten Feldversuch wurde der Mittelwert jeder Signal-Capture in den Merkmal-Vektor eintragen und mit dem zugehörigen Bereich verknüpft. Da pro Raspberry 360 Signal-Captures entstanden sind, enthält der Trainingsdatensatz für die SVM 360 Zeilen mit je vier Spalten. Mit einem solchen Datensatz lässt sich die Support Vector Machine anlernen.

In der Online-Phase des ersten Feldversuchs ging eine Person, bei der sich der Beacon auf Kopfhöhe befand, verschiedene Bereiche des Raumes ab. Über die eingebaute Visualisierung erfolgte die Anzeige der Position mithilfe eines Markers direkt auf der Karte im Browser (vgl. Abbildung 4.1). Dadurch kann die ermittelte Position direkt mit dem korrekten Standort der laufenden Person abgeglichen werden.

Mit einem Capture-Intervall von 500 ms konnte eine Position jedoch nur ansatzweise korrekt bestimmt werden. In den seltensten Fällen wurde selbst im Stillstand die richtige Position über mehr als zwei Capture-Intervalle hintereinander ermittelt. Dadurch konnte lediglich ein Flackern der Positionsmarkierung auf der Karte beobachtet werden, selbst wenn eine Person ohne Bewegung in einem Eckbereich direkt vor einem Empfänger stand.

Der Radius des Flackerns – der Abstand zwischen zwei Bereichen, die durch aufeinander folgende Signal-Captures ermittelt wurden – war unabhängig vom Standort der Person. Aus diesen Beobachtungen lässt sich schließen, dass die Genauigkeit der bisherigen Lokalisierung zur Anwendung mit beweglichen Sendern und stationären Empfängern zu ungenau ist. Dies liegt an stärkeren Schwankungen im Signal und an der Ausrichtung des Senders, die sich häufiger ändert, wenn der Sender sich mit einer Person bewegt, und das Signal stark beeinflusst. Aus diesem Grund muss die Lokisierungsleistung deutlich verbessert werden, um den Ansprüchen zu genügen.



## 4 Optimierung und Ergebnisse

In diesem Kapitel wird auf die verschiedenen Möglichkeiten eingegangen, welche Transformationen auf Signal-Captures angewandt werden können, um Merkmal-Vektoren zu erzeugen. Es wird untersucht welche statistischen Eigenschaften zu den besten Lokalisierungsergebnissen führen. Dazu wird eine Testmethodik eingesetzt, die sowohl messbare, als auch nicht messbare Ergebnisse berücksichtigt. Weiterhin wird die Übertragbarkeit der Ergebnisse auf das klassische Szenario durch die eingesetzte Methodik gewährleistet.

### 4.1 Konzeption einer Testmethodik

Um die Ortungsergebnisse vergleichen und optimieren zu können, wird zuerst eine Testmethodik benötigt, die einerseits messbare Ergebnisse liefert und andererseits beide Szenarien abdeckt. Nur so kann gewährleistet werden, dass Ergebnisse objektiv verbessert werden können, da das Flackern der Markierung auf der Karte in einem Feldversuch nicht reproduzierbar ist. Dennoch ist auch ein Feldversuch zur Beurteilung der Ergebnisse nötig, da eine künstliche Umgebung nie alle Einflüsse der realen Umgebung abbilden kann. Dazu werden in diesem Abschnitt mehrere Tests ausgearbeitet, um vergleichbare und reproduzierbare Ergebnisse zu erhalten.

#### 4.1.1 Vergleichstest für stationär installierte Beacons

Eine Möglichkeit zur messbaren Bewertung der Positionsbestimmung liefert ein bereits vorhandener Integrationstest mit Daten, die auf unbekannte Art und Weise erfasst wurden. Der Test basiert auf Trainings- und Positionsdaten aus [Tre15] zu stationär installierten Beacons mit beweglichen Empfängern. Dabei wurde nicht nur der Raum 01c04 genutzt, sondern zusätzlich der Raum 01c03 mit dem zugehörigen Flurabschnitt (*Hallway*) einbezogen.

Dazu wurden zu allen benannten Bereichen in Abbildung 4.1 Trainingsdaten erhoben. Das Capture-Intervall liegt bei drei Sekunden und die Dauer der Erhebung von Trainingsdaten für ein einzelnes Gebiet ist mit 20 Sekunden angegeben [Tre15]. Die Sendefrequenz der Beacons ist nicht bekannt. Allerdings kann anhand der Daten auf eine Frequenz zwischen 5 Hz und 10 Hz geschlossen werden. Neben den Trainingsdaten wird in dem Test eine Signal-Capture aus dem Bereich 01c04\_02 verwendet.

Der zu ermittelnde Bereich wird mit einer blauen Markierung hervorgehoben. Die Einteilung der Bereiche ist in etwa dieselbe wie im ersten Feldversuch (vgl. rote Linien in Abbildung 4.2). Die Tische hatten jedoch zum Zeitpunkt der Datenhebung für diesen Test eine andere Position.

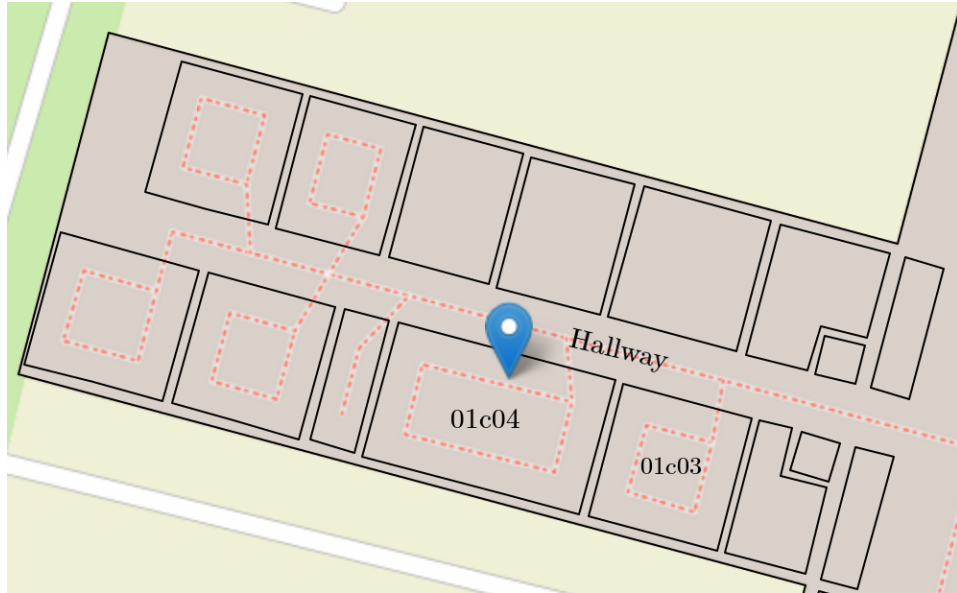


Abbildung 4.1: Die Abbildung stellt die Visualisierung der Testgeometrie, wie sie in der Webanwendung zu sehen ist, dar. Die blaue Markierung zeigt den Bereich 01c04\_02, der im Integrationstest gesucht wird.

Ziel dieses Tests war bisher, die Signal-Capture aus dem Gebiet 01c04\_02 genau diesem zuzuordnen. Allerdings darf dies nicht die einzige Anforderung an das Testergebnis sein, da eine richtige Zuordnung nach einer Optimierung auch Zufall sein könnte. Da der *Location-Solver* bei Klassifikation jeder Signal-Capture eine Liste liefert, die nach Bereich und Konfidenz sortiert ist, lässt sich für jeden Bereich sagen, wie sicher der Klassifikator ist, dass die Signal-Capture aus einem bestimmten Bereich stammt.

Die Konfidenz kann dabei als Maß zur Güte der Positionsbestimmung herangezogen werden und ist durch Definition 4.1.1 definiert. Die Konfidenz ist eine Maßzahl zwischen null und eins. Je größer die Konfidenz ist, desto eher stammt die Signal-Capture aus dem zugehörigen Bereich. Die Summe über alle Konfidenzen aus einer Ergebnisliste ist immer genau eins, da sicher ist, dass die Signal-Capture aus einem der Gebiete der Liste stammt.

**Definition 4.1.1** Für die Konfidenz des  $i$ -ten Bereichs  $K_i$  aus den Ortungsergebnissen einer Signal-Capture gilt:

1.  $K_i \in [0, 1]$
2.  $\sum_{i=1}^n K_i = 1$ , wobei  $n = \text{Anzahl der Gebiete}$

Daher kann die Konfidenz als diskrete Wahrscheinlichkeitsfunktion der Gebietszuordnung betrachtet werden. Eine solche Ergebnisliste ist beispielsweise in den Tabellen 4.1 und 4.2 zu sehen. Diese zeigen die Ergebnisse des Integrationstest vor jeglichen Optimierungen. Aus der Konfidenz als Gütekriterium der Bereichsklassifikation müssen zwei Eigenschaften einer Optimierung gefordert werden:

**Anforderung 4.1.1**

1. *Der Bereich 01c04\_02 muss die höchste Konfidenz besitzen.*
2. *Der Abstand der Konfidenzen zwischen 01c04\_02 und dem Bereich an zweiter Position muss so groß wie möglich sein. Mindestens jedoch 0,3. Es gilt: je größer der Abstand, desto besser die Optimierung.*

Über diese beiden Eigenschaften lässt sich abschätzen, ob eine Optimierung in der Praxis funktionieren kann. Sollten sie nicht erfüllt sein, ist davon auszugehen, dass die vorgenommene Änderung zu keiner Verbesserung im Test und in einem Feldversuch mit beweglichen Beacons führen kann, da die Lokalisierung beweglicher Beacons deutlich anfälliger für Störungen ist. Die Ergebnisse dieses Tests sind daher nur als erster Hinweis zur verbesserten Ortung beweglicher Sender zu werten, lässt jedoch direkte Schlüsse auf die Lokalisierungsleistung stationärer Sender mit beweglichen Empfängern zu.

**4.1.2 Feldversuche für bewegliche Beacons**

Da der Integrationstest alleine keine hinreichende Bewertung für die Ortung beweglicher Sender liefern kann, sollte nach einer erfolgreichen Optimierung im Integrationstest für das klassische Szenario immer ein Feldversuch mit beweglichen Sendern folgen, auch wenn die Ergebnisse nicht direkt messbar sind, da zwei Feldversuche nie identisch ablaufen.

Der Vorteil eines Feldversuchs liegt darin, dass Effekte auftreten können, die in einem einfachen Integrationstest nicht berücksichtigt werden, wie zum Beispiel die Bewegung von anderen Personen im Raum oder unterschiedliche Bewegungsgeschwindigkeiten. Daher lassen sich bei einem Feldversuch mögliche Probleme besser erkennen als in einem programmierten Test. Dennoch wird das Flackern dadurch nicht objektiv messbarer. Es können somit nur Vermutungen über den möglichen Erfolg angestellt werden.

Um insgesamt ein messbares Ergebnis zu erhalten, wird der Feldversuch mit einem weiteren künstlichen Testszenario verglichen werden. Dazu muss zunächst festgelegt werden, welcher Capture-Intervall im Feldversuch und im Test verwendet werden soll. Da im ersten Feldversuch ein Capture-Intervall von nur 500 ms verwendet wurde und davon auszugehen ist, dass ein längerer Capture-Intervall zu einer besseren Positionsbestimmung führt, muss zunächst bestimmt werden, welcher Zeitraum für eine Live-Lokalisierung noch akzeptabel ist.

Menschen bewegen sich in der Regel mit ca. 1,3 m/s. Unter dieser Annahme ist ein Capture-Intervall von 700 ms gerade noch ausreichend, um eine Person auch während einer Bewegung orten zu können, da die Abtastrate in etwa der Nyquist-Frequenz entspricht.

Ein weiterer Grund für diese Grenze ist die von Menschen empfundene Trägheit eines Systems mit einer Reaktionszeit von über 500 ms. Je weiter diese Grenze überschritten wird, desto träger wirkt die Lokalisierung. Nach Erfahrungen des Autors und verschiedenen Versuchspersonen in weiteren Feldversuchen, ist ein Capture-Intervall von 700 ms als Grenze festgestellt worden.

Aus diesem Grund werden alle folgenden Feldversuche mit beweglichen Beacons mit einem Capture-Intervall von 700 ms durchgeführt. Des Weiteren wird die Datenerhebung generalisiert, sodass für alle Feldversuche der gleiche Trainingsdatensatz verwendet wird. Der Unterschied im Vorgehen in den Feldversuchen ist lediglich die Art der Merkmal-Vektor-Erstellung, da auf dieser alle folgenden Optimierungen basieren. Damit lässt sich ein direkter Vergleich zwischen den einzelnen Optimierungen ziehen. Als subjektives Maß gilt die Schwere des Flackerns und dessen Radius.

Neben den Feldversuchen mit beweglichen Beacons werden Feldversuche mit beweglichen Empfängern und stationären Beacons durchgeführt. Dies dient vor allem der Erkennung von Raumwechseln, da Versuche mit beweglichen Beacons ausschließlich im Raum 01c04 bzw. kombiniert mit dem zugehörigen Türbereich des Flurs durchgeführt werden konnten.

### 4.1.3 Praxistest für bewegliche Beacons

Um neben subjektiven Bewertungen in Feldversuchen auch auf objektive Metriken zugreifen zu können, wird ein Testfall benötigt, der sich zum einen direkt mit den Feldversuchen vergleichen lässt. Zum anderen soll er mit gleichen Daten stets zu wiederholbaren Ergebnissen führen, was bei einem Feldversuch nicht garantiert ist.

Damit eine Vergleichbarkeit zu den Feldversuchen gewährleistet ist, wird in den Versuchen und im Testfall derselbe Trainingsdatensatz verwendet. Da, wie zuvor beschrieben, ein Capture-Intervall von 700 ms für alle zukünftigen Tests und Versuche gewählt wurde, muss ein neuer Trainingsdatensatz des Raumes 01c04 mit entsprechenden Signal-Captures angelegt werden. Dazu werden die Raspberries wie im ersten Versuch (vgl. Abbildung 3.6) in den Ecken des Raumes platziert.

Wie in der Abbildung zu erkennen ist, nehmen Tische die Gebiete 01c04\_04 und 01c04\_05 zum Großteil ein. Daher können in diesen Gebieten nur Daten aus den Randbereiche erhoben werden. Zur Erhebung des Trainingsdatensatzes wurde jedes Gebiet für 30 Sekunden mit dem Beacon auf Kopfhöhe – genau wie im ersten Feldversuch – abgegangen. Der so gewonnene Datensatz wird im Folgenden sowohl für Feldversuche als auch Integrationstests eingesetzt. Dadurch ist eine direkte Übertragbarkeit der Ergebnisse auf mögliche Beobachtungen im Feldversuch gewährleistet.

Damit für Integrationstests ein Datensatz mit Referenzpositionen zur Verfügung steht, wurde der Raum, wie in Abbildung 4.2, gerastert. An jedem einzelnen grünen Punkt wurde über 30 Sekunden ein Datensatz erhoben. Die datenerhebende Person musste währenddessen durchgehend in Bewegung bleiben, damit die Signale durch diese verfälscht wurden. Ein Stillstand hätte dazu geführt, dass die Daten Störungen, die durch Bewegungen entstehen, nicht widerspiegeln. Dieses Vorgehen dient der größeren Realitätsnähe. Dadurch entstand ein Positionsdatensatz mit ca. 1200 Signal-Captures mit den zugehörigen Bereichen.

Mithilfe dieser Daten lassen sich die Ergebnisse der Optimierungen messbar bewerten, indem geprüft wird, wie viele Signal-Captures aus einem Bereich diesem von der Support Vector Machine tatsächlich zugeordnet werden. Eine gelungene Optimierung sollte daher die folgenden Eigenschaften im Referenztest aufweisen:

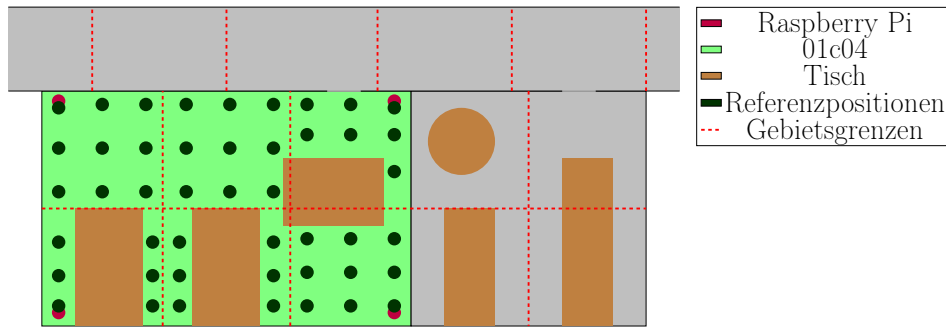


Abbildung 4.2: Der Versuchsaufbau ist identisch zu dem aus Abbildung 3.6. Zusätzlich enthalten sind die dunkelgrün markierten Positionen, an denen die Referenzdaten erhoben wurden.

#### Anforderung 4.1.2

1. Die *Signal-Captures* müssen dem korrekten Bereich mit der höchsten relativen Häufigkeit zugeordnet werden.
2. Die relative Häufigkeit sollte über 80% liegen.

Durch diese hohen Anforderungen entsteht die Möglichkeit problematische Bereiche zu identifizieren und weitere Verbesserungen vorzunehmen. Des weiteren ist zu erwarten, dass das Flackern bei einer zu 80% richtigen Zuordnung der Bereiche nur noch in geringem Maße auftritt. Dieser Test wird aufgrund der Klassifikation von originalen Positionsdaten im weiteren Verlauf *Praxistest* genannt.

## 4.2 Modelloptimierung

Um eine Optimierung möglich zu machen, muss zunächst das bisher eingesetzte Grundmodell analysiert werden. Dabei wird insbesondere darauf eingegangen, wie mit ausbleibenden Signalen umgegangen wird und welcher Einfluss auf den Merkmalraum dadurch entsteht. Als Optimierungsmaß wird der Integrationstest zugrunde gelegt.

### 4.2.1 Bisheriges Modell

Der wichtigste Aspekt einer Lokalisierung mit einem Fingerprintingsystem besteht in der Zuordnungsgenauigkeit der Live-Signale zu den in der Offline-Phase aufgenommenen Fingerabdrücken. Da das Fingerprinting in dieser Arbeit auf einer Support Vector Machine basiert, müssen die Merkmal-Vektoren so angelegt sein, dass sie im Merkmalraum möglichst gehäuft auftreten und die Haufen unterschiedlicher Klassen wenige gemeinsame Punkte besitzen.

Dies kann mit beliebigen Verfahren erfolgen. Bisher werden alle RSSI-Werte einer Signal-Capture bzw. in einem Capture-Intervall gemittelt und als Merkmal-Vektor-Eintrag

benutzt. Allerdings handelt es sich dabei um den Standardfall, dass alle Empfänger jedes Signal empfangen haben.

Genauer betrachtet werden muss zunächst der Sonderfall, in dem ein Signal verschwindet, wie zum Beispiel durch Auslöschung oder Absorption, und der Fall, dass die Reichweite des Senders nicht ausreichend ist, damit ein Empfänger das Signal verarbeiten kann. Dadurch, dass Beacons nur Broadcasting-Signale aussenden, ist zu keinem Zeitpunkt klar, ob ein einzelnes Signal auf dem Weg verschwunden ist oder ob sich der Empfänger nicht in der Reichweite des Senders befindet.

Bei einer einfachen Mittelwertbildung werden nur die Signale aus dem zuletzt empfangenen Capture-Intervall zur Erzeugung eines Eintrags im Merkmal-Vektor genutzt. Wenn das Signal verloren wurde oder der Empfänger außerhalb der Reichweite steht, wird im Merkmal-Vektor ein Eintrag mit dem Wert Null generiert.

Wird ein gleitendes Fenster benutzt, wird für den Merkmal-Vektor-Eintrag der Mittelwert aller RSSI-Werte der letzten Capture-Intervalle verwendet. Bei einem Fenster von drei Capture-Intervallen werden alle RSSI-Werte der letzten drei Intervalle gemittelt. Tritt im Verlauf der Messung ein Signalverlust über ein Intervall auf, sind keine RSSI-Werte für das letzte Capture-Intervall vorhanden.

Dadurch werden keine weiteren RSSI-Werte zur Mittelung hinzugezogen. Das hat zur Folge, dass unter bestimmten Umständen ein Paketverlust nicht zu einer Veränderung des Mittelwertes führt. Bei einem Fenster von drei Capture-Intervallen träte dieser Fall ein, wenn zum Beispiel zuerst ein RSSI-Wert in einem Intervall vorhanden ist und in den folgenden beiden Intervallen nicht.

Dann bliebe der Mittelwert über drei Intervalle konstant. Ein Ausbleiben des Signals würde in keiner Weise bestraft. Beim vierten Intervall ohne Eingangssignal wird der Empfänger als außerhalb der Reichweite gewertet und somit eine Null in den Vektor eingetragen. Dieses Vorgehen führt im zeitlichen Verlauf betrachtet zu abrupten Änderungen.

Gleichzeitig entsteht dadurch ein Wertebereich, der physikalisch nicht abgedeckt wird, worauf im nächsten Absatz (4.2.2) genauer eingegangen wird. Diese ursprüngliche Vorgehensweise soll anhand des zuvor beschriebenen Integrationstests analysiert und beurteilt werden.

In Tabelle 4.1 sind die ersten fünf Ergebnisse der Klassifikation aus dem Integrationstest dargestellt. Der Bereich 01c04\_02 wurde mit der Konfidenz von 0,39 als am Wahrscheinlichsten bewertet. Dieses Ergebnis stimmt mit der Erwartung an den Test überein, da auf ein Signal aus diesem Bereich getestet wurde.

Bemerkenswert an dieser Tabelle ist jedoch der geringe Abstand von 0,06 der Konfidenzen der Bereiche 01c04\_02 und 01c04\_04, der in Abbildung 4.1 im Raum 01c04 in der linken unteren Ecke im Schnittpunkt der roten Linien zu finden ist. Das lässt darauf schließen, dass die Standortfindung in diesem Fall funktioniert hat, jedoch in der Praxis stark flackert, was ebenfalls bei den ersten Versuchen in Abschnitt 4.1 beobachtet werden konnte. Daher ist das zweite Kriterium aus Anforderung 4.1.1 nicht erfüllt.

Tabelle 4.2 zeigt die Ergebnisse der Klassifikation mit gleitendem Mittelwert über zwei Capture-Intervalle. Besonders auffällig ist, dass der erwartete Bereich 01c04\_02 nicht in der Tabelle enthalten ist und daher eine Konfidenz unter 0,06 besitzt. Stattdessen wird der Bereich 01c04\_01, der in Abbildung 4.1 direkt links neben dem Bereich 01c04\_02



Bereich	Konfidenz
01c04_02	0,39
01c04_04	0,33
01c04_01	0,10
01c04_03	0,03
01c03_02	0,02
...	

Tabelle 4.1: Ergebnisse des Integrationstests mit einfachem Mittelwert

Bereich	Konfidenz
01c04_01	0,26
hallway_01	0,11
01c04_03	0,11
hallway_02	0,08
01c04_04	0,06
...	

Tabelle 4.2: Ergebnisse des Integrationstests mit gleitendem Mittelwert

liegt, als am Wahrscheinlichsten gewertet. Während in 4.1 vier von sechs Bereichen aus dem Raum 01c04 gut bewertet wurden, schwankt die Bewertung in Tabelle 4.2 zwischen Flur und Raum 01c04.

Allerdings ist der Abstand der beiden höchsten Konfidenzen mit 0,15 größer als beim einfachen Mittelwert. Da jedoch bereits die erste Forderung an den Test nicht erfüllt werden konnte, ist der Integrationstest mit gleitendem Mittelwert fehlgeschlagen. Eine Optimierung des Umgangs mit nicht empfangenen Signalen soll dazu führen, dass sowohl der Test mit dem einfachen Mittelwert, als auch der Test mit gleitendem Mittelwert die Kriterien aus Anforderung 4.1.1 vollständig erfüllt.

## 4.2.2 Optimierung

### Vorüberlegungen

Um den Aufbau des Merkmalraums grundsätzlich zu verbessern, sollte der Umgang mit verloren Signalen in den Merkmal-Vektoren verbessert werden, da hier die einzige Möglichkeit besteht direkten Einfluss zu nehmen, wenn Transformationen zunächst außer Acht gelassen werden. Des Weiteren werden bisher keine physikalischen Grenzen berücksichtigt. Wie in Abschnitt 2.2 bereits beschrieben liegt der Wertebereich der RSSI-Werte bei ca.  $[-100, -30]$ , wobei  $-30$  dBm ein sehr nahes Signal beschreibt, während bei einem Wert von  $-100$  dBm das Signal nahezu nicht mehr zu empfangen ist.

In einem kleinen Experiment, in dem ein Android-Tablet im absoluten Grenzbereich der Signalreichweite bewegt wurde, konnte festgestellt werden, dass in seltenen Fällen auch ein Wert von  $-100$  dBm geringfügig unterschritten wird. Dabei wurde ein Minimum von  $-102$  dBm gemessen. Eine Signalstärke von  $-30$  dBm wurde jedoch nie überschritten. Auch nicht, wenn der Empfänger direkt neben dem Sender positioniert wurde.

Eine Null in den Merkmal-Vektor einzufügen, wenn kein Signal empfangen wurde, oder den Mittelwert im Falle eines gleitenden Fensters konstant zu lassen und nicht zu senken, führt logisch betrachtet zu einem verbesserten Eintrag im Merkmal-Vektor, obwohl ein nicht empfangenes Signal zu einer Verschlechterung führen sollte.

Eine bessere Lokalisierung bedeutet gleichzeitig, dass die Merkmal-Vektoren der zu differenzierenden Bereiche sich deutlich genug voneinander unterscheiden. Dazu müssen die Werte in den Merkmal-Vektoren für jeden einzelnen Bereich möglichst dicht gehäuft sein.

Abbildung 4.3 zeigt den bisherigen Wertebereich der Merkmal-Vektoren und die daraus entstehenden Gruppierungen mit zugehörigen Bereichen der Merkmal-Vektoren von zwei Beacons. Für die Erstellung der Merkmal-Vektoren wurde ein einfaches arithmetisches Mittel verwendet. Grau hinterlegt ist der physikalisch mögliche Messbereich, in dem sich die tatsächlich gemessenen RSSI-Werte befinden. Orange hinterlegt ist der Wertebereich, der durch die künstlich eingefügten Nullen entsteht. Werden die Werte in den orangenen Flächen nicht berücksichtigt, ist die Gruppierung der Werte zu einzelnen Bereichen deutlich zu sehen.

Zu beachten ist vor allem die Überlappung der blauen und roten Bereiche, die auch in Tabelle 4.1 sehr dicht beieinander liegen. Die künstlich eingefügten RSSI-Werte stören allerdings die kompakte Gruppierung der Messwerte im physikalisch möglichen Bereich. Dadurch wird die Gruppierung auseinander gezogen und die Überlappung der Werte auch außerhalb der Gruppen fortgeführt. So lassen sich die Gruppen schlecht algorithmisch und visuell voneinander trennen.

Besser ist es daher, die künstlich eingefügten RSSI-Werte an den physikalischen Wer-

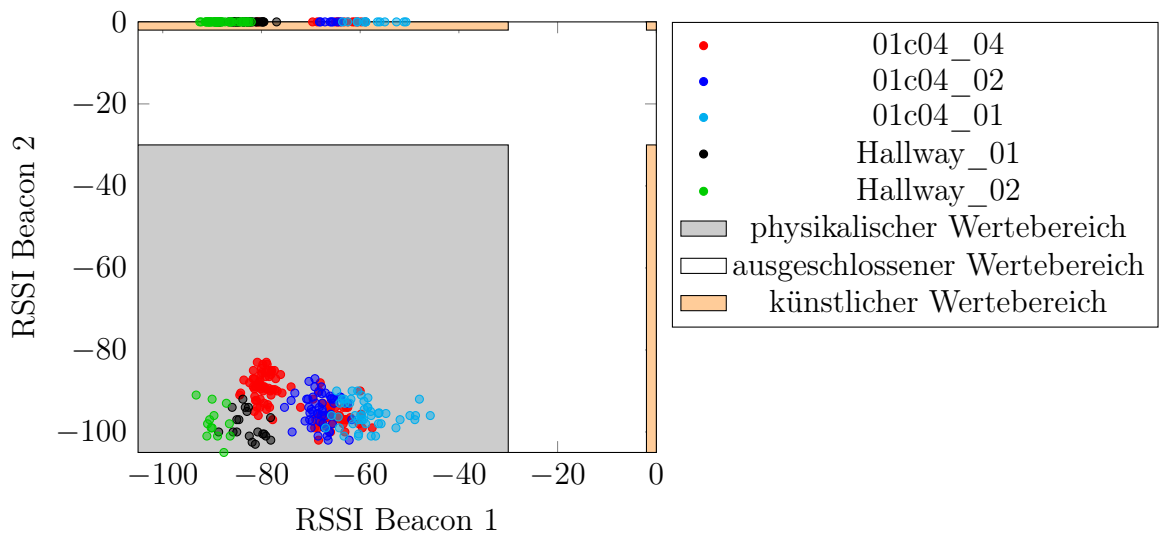


Abbildung 4.3: Die Abbildung zeigt den bisherigen Wertebereich des Merkmalraumes. Grau hinterlegt ist der physikalisch mögliche Wertebereich. Die künstlich eingefügten Ränder liegen orange unterlegt außerhalb des grauen Bereichs. Die eingefärbten Punkte stellen pro Farbe einen Raumbereich dar. Eine günstigere Gruppierung liegt vor, je näher gleichfarbige Punkte beieinander liegen, ohne sich mit anders gefärbten Punkten zu überlappen. Durch die künstlich eingefügten Werte entsteht eine deutliche Lücke zum physikalisch möglichen Wertebereich

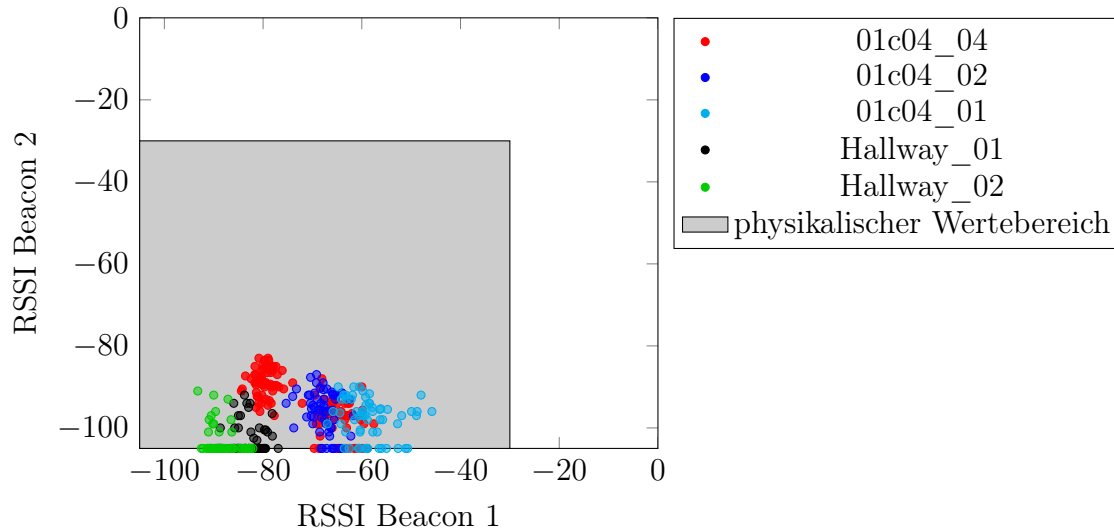


Abbildung 4.4: Zu sehen ist der optimierte Wertebereich des Merkmalraums. Hier sind ausschließlich Werte annehmbar, die physikalisch empfangen werden können. Dadurch liegen gleichfarbige Punkte bei ausbleibenden Signalen näher beieinander und die Gruppierungen sind besser zu unterscheiden.

tebereich anzupassen. Zum Beispiel durch das Einfügen des minimal messbaren Wertes bei einem ausbleibenden Signal. Die Merkmal-Vektoren eines solchen Modells sind in Abbildung 4.4 zu sehen. Dabei wird deutlich, dass sich alle Werte im physikalisch möglichen Bereich befinden und durch bloßes Hinsehen eindeutige Gruppen erkennbar sind. Dadurch wird die Klassifikation der Bereiche auch für Algorithmen einfacher, obwohl sich an der grundsätzlichen Überlappung der blauen und roten Bereiche nichts geändert hat.

### Umsetzung

Aus den vorherigen Messungen und Überlegungen lässt sich der Wertebereich der empfangenen Signalstärken auf  $[-105, -30]$  festlegen. Dadurch ergibt sich zwischen dem gemessenen Minimum von  $-102$  dBm ein minimaler Puffer. Sollte innerhalb eines Capture-Intervalls kein Signal empfangen werden, egal ob durch einen einfachen Verlust, oder weil sich der Empfänger außerhalb der Reichweite des Senders befindet, wird ein künstliches Signal mit der Stärke  $-105$  dBm in den Capture-Intervall eingefügt.

Dadurch werden sowohl die Merkmal-Vektor-Erstellung ohne gleitendem Fenster, als auch die Erstellung mit einem gleitendem Fenster direkt beeinflusst. Dies führt zu einer abrupten Bestrafung, genau wie zuvor, eines leeren Capture-Intervalls, indem der Wert auf  $-105$  gesetzt wird. Im Gegensatz zum ursprünglichen Ansatz werden zusätzlich noch die physikalischen Grenzen berücksichtigt. Bei einer Klassifikation mit gleitendem Fenster führt die Einführung dieses künstlichen Signals zu einem langsamen Abfall des Mittelwertes, bis schließlich ein Wert von  $-105$  erreicht wird.

Dies steht im Gegensatz zu der bisherigen Vorgehensweise, bei der ein einzelnes leeres

Bereich	Konfidenz
01c04_02	0,60
01c04_04	0,24
01c04_01	0,09
01c04_05	0,03
01c04_06	0,01
01c04_03	0,01
...	

Tabelle 4.3: Ergebnisse mit optimiertem Modell und einfachem Mittelwert

Bereich	Konfidenz
01c04_02	0,59
01c04_04	0,16
01c04_01	0,16
01c04_05	0,04
01c04_06	0,01
01c04_03	0,01
...	

Tabelle 4.4: Ergebnisse mit optimiertem Modell und gleitendem Fenster

Capture-Intervall den Mittelwert wenig bis gar nicht beeinflusst hat und erst bei mehreren Capture-Intervallen ohne empfangenes Signal abrupt auf Null gesetzt wurde, was physikalisch die beste, erreichbare Signalstärke ist.

### Integrations- und Praxistest

Wird der Integrationstest aus dem vorherigen Abschnitt mit dem optimierten Modell durchgeführt, ergeben sich die Konfidenzwerte aus den Tabellen 4.3 und 4.4, die jeweils die Konfidenz der sechs am besten bewerteten Bereiche wiedergeben. An diesen Tabellen fallen zunächst zwei Dinge auf: Erstens liefern beide Klassifikationen den erwarteten Bereich 01c04\_02 als Bereich der höchsten Konfidenz. Zweitens ist die Konfidenz für diesen Bereich drastisch gestiegen.

Zunächst sollen die Ergebnisse der Optimierung ohne gleitenden Mittelwert betrachtet werden. Diese werden Tabelle 4.3 dargestellt. Dabei ist zu erkennen, dass durch die Modelloptimierung eine Konfidenz von 0,60 im Vergleich zu einer vorherigen Konfidenz von 0,39 erreicht wird.

Dies bedeutet eine Steigerung von 0,21 bei einer gleichzeitigen Senkung der Konfidenzen aller anderen Bereiche. Des Weiteren geht aus dieser Tabelle hervor, dass sich nach der Optimierung alle sechs Bereiche aus Raum 01c04 unter den sechs Bereichen der höchsten Konfidenz befinden.

Weiterhin wurde der Abstand zwischen den beiden am besten bewerteten Bereichen von 0,06 auf 0,36 gesteigert. Beide zuvor gestellten Anforderungen an eine Optimierung wurden bei der Klassifikation ohne gleitenden Mittelwert daher vollständig erfüllt.

Bei der Betrachtung der Ergebnisse unter Verwendung eines gleitenden Mittelwertes in Tabelle 4.4 fällt auf den ersten Blick auf, dass sich viele Bereiche unter den wahrscheinlichsten sechs befinden, die vor der Optimierung nicht darunter waren. Außerdem finden sich ebenfalls alle sechs Bereiche aus Raum 01c04 unter den wahrscheinlichsten sechs wieder.

Wie beim einfachen Mittelwert ist der Bereich 01c04\_02 nun der Bereich mit der höchsten Konfidenz, wie im Testfall gefordert. Seine Konfidenz wurde von 0,01 auf 0,59

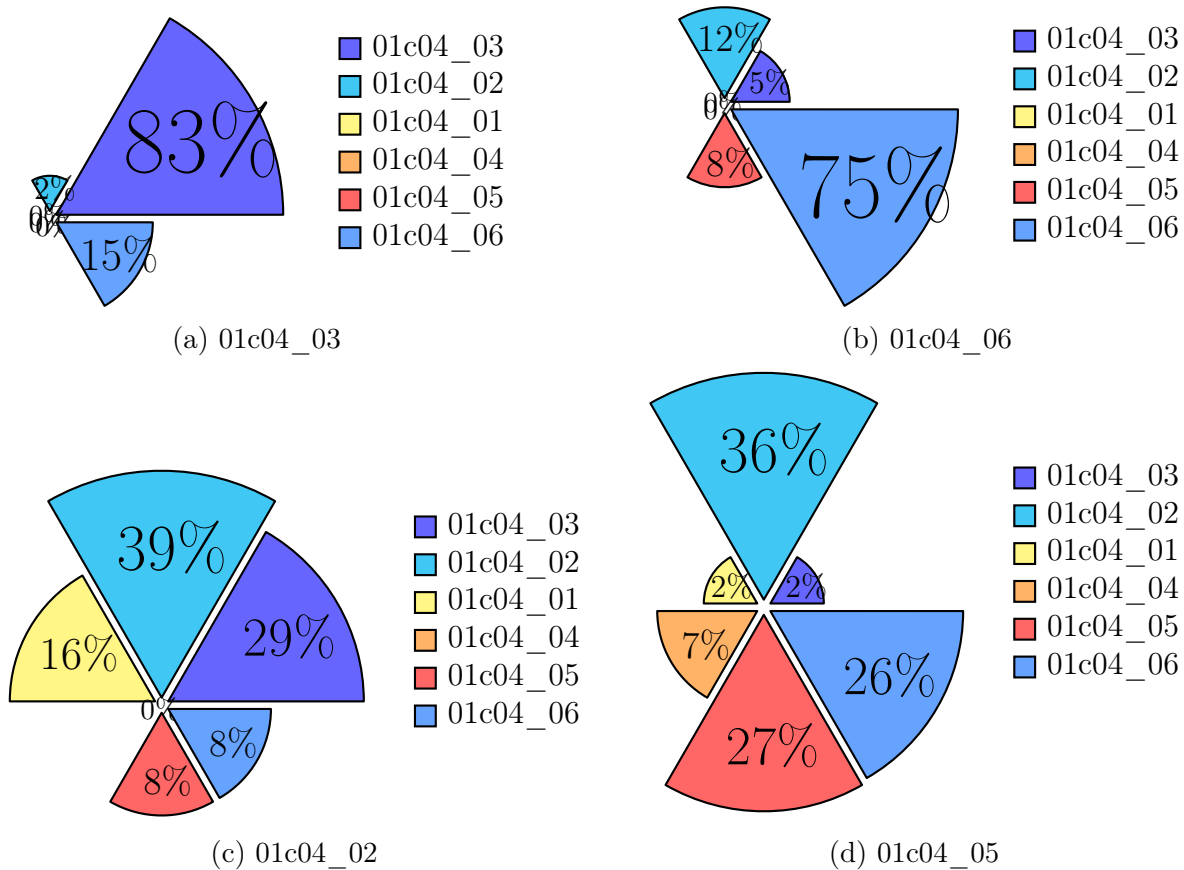


Abbildung 4.5: Abgebildet sind die relativen Häufigkeiten der ermittelten Bereichszuordnungen aus dem Praxistest über den Mittelwert als genutztes Transformationsmerkmal. Den Bildunterschriften ist zu entnehmen, aus welchem Bereich die zu klassifizierenden Signal-Captures des Positionsdatensatzes stammen. Das Ergebnis ist besser je höher der Anteil der Zuordnungen zum gleichnamigem Bereich der Bildunterschrift ist. Die einzelnen Tortenstücke sind analog zu den Raumbereichen angeordnet.

gesteigert. Neben dieser Steigerung wurde auch der Abstand zum Bereich mit der zweithöchsten Konfidenz (01c04\_04 in Tab. 4.4) von 0,15 auf 0,43 erhöht.

Den Ergebnissen des Integrationstests zufolge, konnte die vorgenommene Optimierung des Modells erheblich bessere, und unter Verwendung eines gleitenden Mittelwertes im Vergleich zu vorher richtige Ergebnisse liefern. Nach den angewandten Kriterien ist das gleitende Mittel nach der Modelloptimierung sogar besser zu bewerten als die Merkmal-Vektor-Erstellung mit einem einfachen Mittelwert.

Zur Überprüfung, ob die Ergebnisse aus dem Integrationstest ebenfalls in der Anwendung mit beweglichen Sendern und stationären Empfängern zu einer besseren Positionsbestimmung führt, müssen ein Feldversuch und der zugehörige Test durchgeführt werden. Nur so kann festgestellt werden, ob das andauernde Flackern zwischen einzelnen Raumbereichen durch die Modellanpassung reduziert wurde. Die Merkmal-Vektoren wurden

in den Überprüfungen mit dem einfachen Mittelwert der Signal-Captures erstellt.

Abbildung 4.5 zeigt die Ortungsergebnisse für ausgewählte Raumabschnitte. Die beiden unteren Abbildungen 4.5a und 4.5b stellen Lokalisierungsergebnisse in Raumecken dar. Die zugehörigen Bereiche besitzen dementsprechend nur zwei Nachbarbereiche. Dadurch ist die Ortung in diesen Bereichen sehr genau. Gleiches gilt auch für die nicht abgebildeten Bereiche 01c04\_01 und 01c04\_04. Sie alle kommen jeweils auf eine Ortungsgenauigkeit von zwischen ca. 75% und 85%, je nach Position der Person bzw. Genauigkeit beim Einmessen.

Weiterhin ist in Abbildung 4.5b auffällig, dass 12% der Signale dem Bereich 01c04\_02 zugeordnet werden, obwohl dieser keine gemeinsamen Grenzen mit 01c04\_06 besitzt, sondern mit dessen Nachbarn. Im Feldversuch ist daher zu erwarten, dass der Radius des Flackerns größer sein wird, als im Bereich 01c04\_03. Dennoch ist in den Eckbereichen mit einem merklich reduzierten Flackern zu rechnen.

Weniger eindeutig fällt die Ortung in den beiden mittleren Bereichen 01c04\_02 (Abbildung 4.5d) und 01c04\_05 (Abbildung 4.5c) aus. Testdaten aus dem Bereich 01c04\_05 werden nur zu 26% auch diesem Bereich zugeordnet, während 36% in den gegenüberliegenden Bereich 01c04\_05 eingeordnet werden. Weitere 26% werden dem rechten Nachbarbereich 01c04\_06 zugerechnet. Dem Bereich 01c04\_02 konnten über den Mittelwert immerhin in 39% der Fälle korrekt zugeordnet werden. Trotzdem ist im Feldversuch mit einem starken Flackern zu rechnen.

Das bedeutet, dass bei einer realen Ortungsanwendung eine Person nur selten dem richtigen Bereich zugeordnet werden kann, wenn sie sich in einem mittleren Raumbereich aufhält. Dies konnte im Feldversuch nachvollzogen werden. An den Rändern konnte das nahezu zufällige Flackern aus den ersten Tests spürbar reduziert werden. Dies ist ein deutliches Anzeichen für eine Verbesserung der Lokalisierung durch den Umgang mit nicht empfangenen Signalen.

Ähnliche Ergebnisse konnten ebenfalls im umgekehrten Fall mit beweglichen Empfängern und stationären Beacons in einem weiteren Feldversuch erzielt werden, sodass diese Optimierung auf beide Szenarien anwendbar ist. Die Optimierung des Umgangs mit verlorenen oder nicht empfangenen Signalen ist daher als erfolgreich zu werten, auch wenn mittlere Raumbereiche nur eine Ortungsgenauigkeit von unter 40% besitzen.

### 4.3 Signalanalyse

Nach der erfolgten Modelloptimierung soll eine Signalanalyse Aufschluss über weitere Optimierungsmöglichkeiten bei der Merkmal-Vektor-Erstellung geben. Dazu werden im Folgenden die Dichten der empirischen Verteilungsfunktion verschiedener Signalmessungen über eine Minute in verschiedenen Abständen ausgewertet.

Abbildung 4.6 zeigt die Dichte im klassischen Anwendungsfall: Ein Beacon ist stationär in einer Ecke des Raumes platziert, während sich der Empfänger im Rauminneren befindet. Die Aufzeichnung der empfangenen RSSI-Werte erfolgte in beiden Abbildung über jeweils eine Minute mit verschiedenen Entfernungen zwischen Sender und Empfänger. Bei Abbildung 4.6 waren Sender und Empfänger direkt aufeinander ausgerichtet. Auf

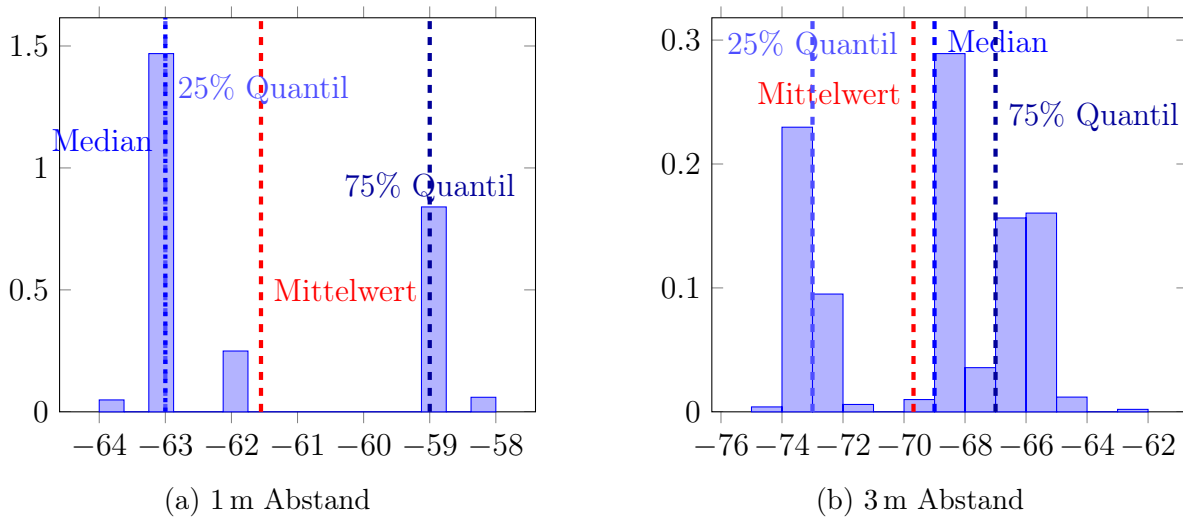


Abbildung 4.6: Dichte-Histogramm der empirischen Verteilungsfunktion von empfangenen RSSI-Werten bei verschiedenen Abständen im klassischen Szenario. Sender befanden sich während des Experiments in einer Ecke und waren direkt auf den Empfänger ausgerichtet. In a sind 25%-Quantil und Median identisch.

den ersten Blick fällt auf, dass die Histogramme nicht der Normalverteilung entsprechen, wie es bei einfachen Messfehlern üblich wäre.

Da der Abstand während der Messung nicht verändert wurde, bleibt nur der Schluss, dass die RSSI-Werte im klassischen Anwendungsfall nicht normalverteilt sind. Was ebenfalls bei Betrachtung der Histogramme auffällt, ist die Übereinstimmung der Quantile mit charakteristischen Positionen. Dies steht im Gegensatz zum Mittelwert, der zwischen den Ausschlägen in den Histogrammen zu finden ist.

Weiterhin ist zu beobachten, dass in beiden Histogrammen das rechte Maximum einen geringeren Wert als die anderen großen Ausschläge besitzen. Dies entspricht nicht dem erwarteten Ergebnis, da kleinere RSSI-Werte den direkten Weg von Sender zu Empfänger widerspiegeln. Im Umkehrschluss bedeutet diese Beobachtung, dass mehr indirekte Signale empfangen wurden als direkte. Eine alternative Erklärung, vor allem für die Dichte in Abbildung 4.6a, wäre eine Schwankung des Signals, erzeugt durch das Beacon selbst, da die Signalzeitreihe hauptsächlich aus zwei Werten besteht.

Um zu untersuchen, ob die RSSI-Verteilung beweglicher Beacons in unterschiedlichen Abständen und Ausrichtungen ähnliche Eigenschaften aufweist wie der klassische Fall, wurden weitere Experimente durchgeführt. Dabei wurde ein Beacon im Rauminnen platziert, während ein Raspberry Pi als Empfänger am Rand positioniert wurde. In dieser Konfiguration wurden die RSSI-Werte eine Minute lang mit Abständen von einem, drei und acht Metern zwischen den Geräten ermittelt.

Dabei waren Sender und Empfänger einmal direkt zueinander ausgerichtet. Beim zweiten Durchlauf wurde der Sender um 90 Grad gedreht, sodass er zur Wand ausgerichtet war. Die zugehörigen Histogramme finden sich in Abbildung 4.7. In der linken Spalte wur-

de das Beacon direkt auf den Sender ausgerichtet. Die rechte Spalte enthält Histogramme bei denen das Beacon um 90 Grad gedreht wurde, um den Einfluss der Ausrichtung auf die Verteilung zu prüfen, da Personen sich normalerweise bei einer Positionsbestimmung bewegen und dadurch nicht immer ideal auf einen Empfänger ausgerichtet sind.

Auch hier fällt auf, dass kein Histogramm einer Normalverteilung nahe kommt. Außerdem unterscheiden sie sich sowohl in Abhängigkeit zur Entfernung, als auch zur Ausrichtung deutlich. Eine kleine Ausnahme bilden die Histogramme im Abstand von drei Metern, bei denen die Ausrichtung einen eher untergeordneten Einfluss auf die Verteilung zu besitzen scheint. Dennoch sind die Unterschiede in der Dichte gut erkennbar.

Sie sind vor allem auf Reflexionen des Signals durch Wände oder den Boden zurückzuführen. Bei Betrachtung der Rohdaten sticht hervor, dass jeweils zwischen 800 und 1000 Signale gemessen wurden, obwohl nur 60 Sekunden bei einer Sendefrequenz von ca. 10 Hz gemessen wurde. Dadurch hätten ca. 600 Signale empfangen werden sollen. Daraus lässt sich folgern, dass zwischen 25% und 40% der gemessenen RSSI-Werte durch Reflexionen entstehen.

Dadurch lässt sich die Periodizität, die in den Dichten scheinbar enthalten ist, erklären. Des Weiteren ist daraus zu schließen, dass Reflexionen zur Verbesserung der Ortungsgenauigkeit möglichst gefunden und mit in die Ortung einbezogen werden sollten.

Genau wie im klassischen Szenario befinden sich die 25%-, 50%- und 75%-Quantile jeweils an signifikanten Stellen der Verteilung, sodass in allen Fällen darauf geschlossen werden kann, dass die Quantile die Verteilung bei der Ortung besser widerspiegeln als der Mittelwert. Eine Ausnahme bildet allerdings Abbildung 4.7b, bei der die Quantile nur innerhalb der reflektierten Signalanteile liegen. Die kleinere Verteilung am rechten Rand wird nicht erfasst, was bei einer Positionsbestimmung zu Problemen führen könnte.

Ebenso problematisch sind die Differenzen zwischen den Quantilen in Abhängigkeit zur Ausrichtung des Beacons. Im Abstand von einem Meter liegen 25%-Quantil und Median dicht beieinander. Das 75%-Quantil unterscheidet sich jedoch erheblich mit  $-62$  im Vergleich zu  $-72$ . Wird die Ausrichtung über verschiedene Abstände miteinander verglichen, ist festzustellen, dass der Median ansatzweise über die Entfernung konstant bleibt, während die anderen Quantile stärkeren Schwankungen unterworfen sind.

Werden zudem Abbildungen 4.7d und 4.7e gegenübergestellt, wird erkennbar, dass auch der Median alleine keine sichere Lokalisierung ermöglichen kann, da der Median in beiden Abbildungen bei  $-72$  liegt, allerdings eine Entfernung von ca. fünf Metern zwischen den beiden Messpunkten liegt.



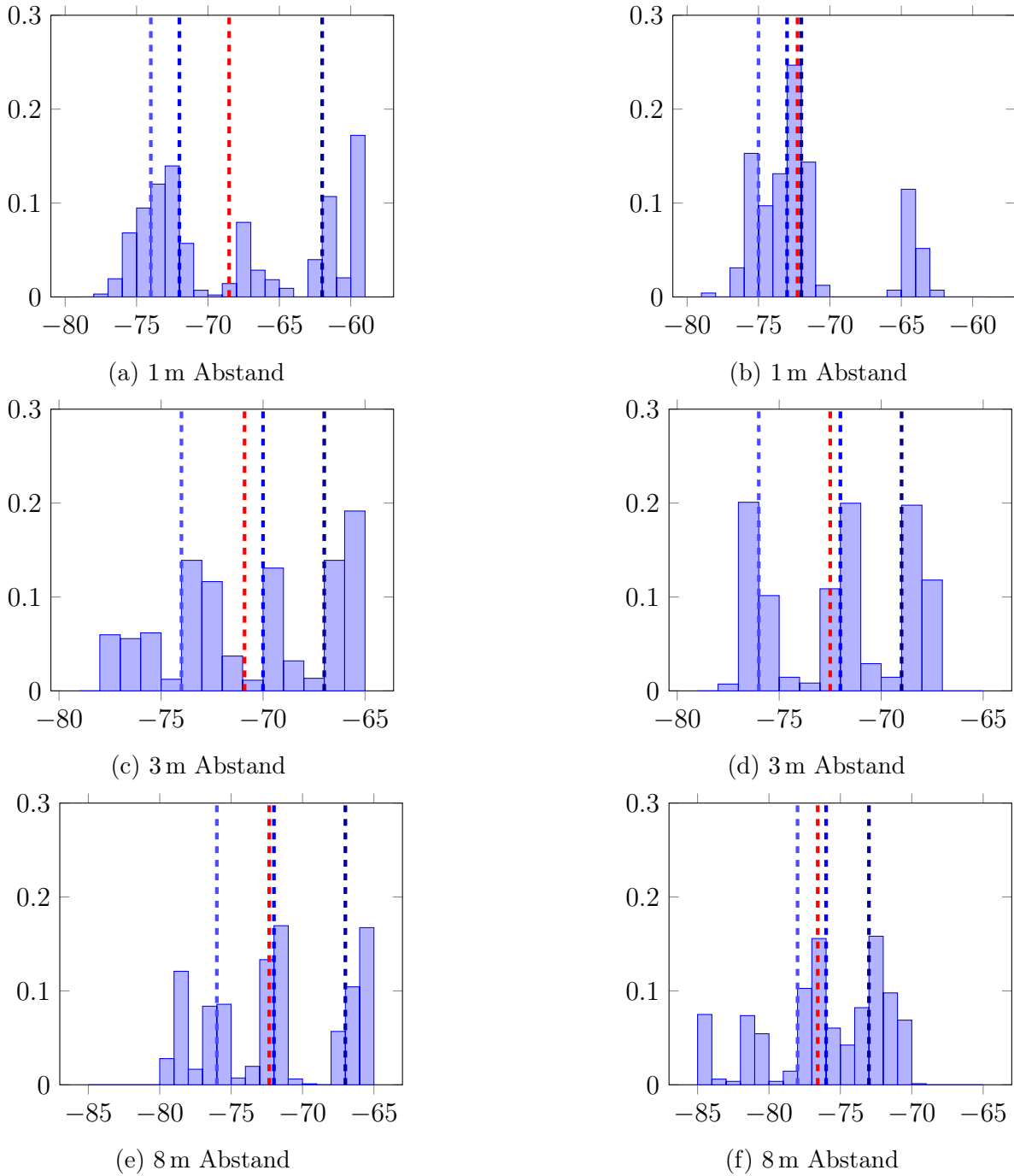


Abbildung 4.7: Die Abbildungen zeigen die Dichtefunktionen der aufgenommenen RSSI-Werte bei verschiedenen Abständen zwischen Sender und Empfänger. Der Sender befand sich in der Raummitte während der Empfänger am Rand des Raumes positioniert wurde. In der linken Spalte wurden sie direkt aufeinander ausgerichtet, in der rechten Spalte wurde der Sender um 90 Grad gedreht. Die rote Linie markiert in den Abbildungen den Mittelwert. Die einzelnen Quantile sind blau markiert, beginnend links in hellblau mit dem 25%-Quantil, dem Median in blau und dem 75%-Quantil in dunkelblau.

## 4.4 Merkmal-Vektor Optimierung

In diesem Absatz werden verschiedene statistische Merkmale einer Signal-Capture untersucht, über die ein Merkmalraum gebildet werden kann. Dazu werden zunächst Vorüberlegungen anhand des Integrationstests angestellt, bevor die Transformation im Praxistest durchgeführt wird. Abschließend erfolgt zu jedem Merkmal ein Feldversuch, um die Ergebnisse aus dem Praxistest in einer realen Umgebung zu verifizieren.

### 4.4.1 Theoretische Überlegungen und Integrationstest für stationäre Sender

Ein typischer Ansatz zur Signalaufbereitung in der digitalen Signalverarbeitung ist die Verwendung von Glättungsfiltern. Diese sind für gewöhnlich erwartungstreu. Das bedeutet, dass sie gegen den Mittelwert einer Messreihe konvergieren. Das hätte allerdings zur Folge, dass die Reflexionen, die signifikant für einen bestimmten Ort sind, herausgeglättet werden. Dadurch würden Informationen zur Verteilung der Signalstärken verloren gehen.

Aus diesem Grund werden im Folgenden der Median und die verschiedenen Quantile auf ihre Tauglichkeit als Merkmal-Vektor untersucht. Dabei soll vor allem auf die Trennbarkeit der Gruppierungen – wie beim Mittelwert – eingegangen werden. Der Integrationstest für das klassische Szenario dient dazu als Messbarkeitskriterium. Daher werden zuerst anhand des Einsatzes von stationären Sendern mögliche Optimierungen erarbeitet und im weiteren Verlauf auf die Ortungsverbesserung beweglicher Sender übertragen.

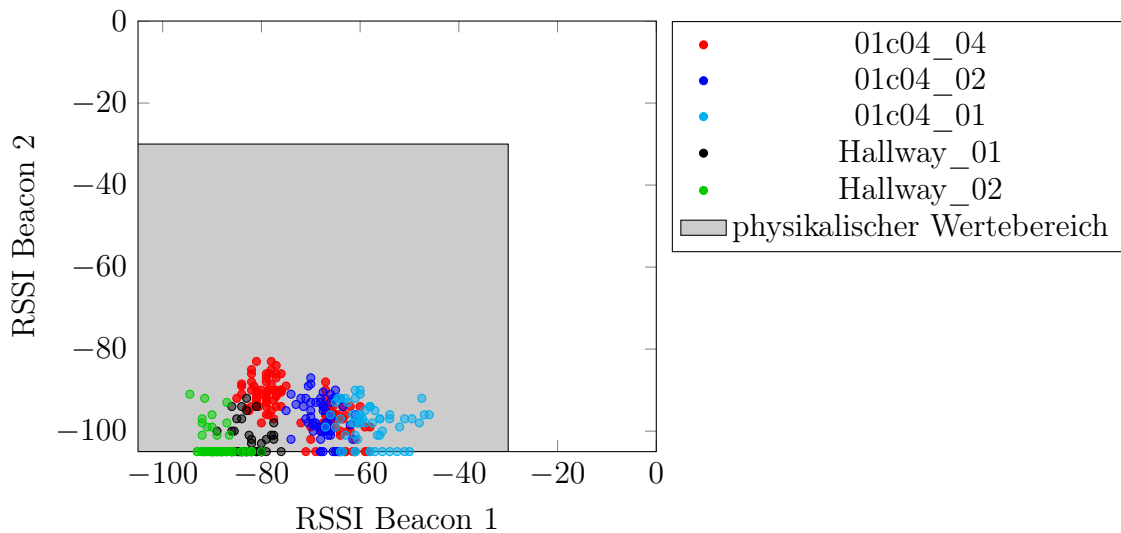


Abbildung 4.8: In der Abbildung ist ein Ausschnitt aus dem Wertebereich des Merkmalraumes zu sehen, der durch die Transformation über den Median der Signal-Captures entstanden ist. Die einzelnen Punkte sind pro Farbe enger gruppiert als beim Mittelwert.

Bereich	Konfidenz
01c04_02	0,77
01c04_01	0,08
01c04_04	0,06
01c04_05	0,03
01c04_06	0,02
01c04_03	0,02
...	

Tabelle 4.5: Klassifikationsergebnisse  
mit einfachem Median

Bereich	Konfidenz
01c04_02	0,66
01c04_01	0,23
01c04_03	0,04
01c04_05	0,02
01c04_04	0,02
01c04_06	0,01
...	

Tabelle 4.6: Klassifikationsergebnisse  
mit gleitendem Median

Dementsprechend gelten die gleichen Voraussetzungen wie in Abschnitt 4.2.1, als der Mittelwert zur Merkmal-Vektor-Erstellung untersucht wurde. Abbildung 4.8 zeigt den Wertebereich, der entsteht, wenn aus jeder Signal-Capture der Median anstelle des Mittelwertes benutzt wird.

Erst bei genauerer Betrachtung ist ein wichtiger Unterschied im Vergleich zum Mittelwert zu erkennen: Die blaue Gruppe des Bereichs 01c04\_02 enthält deutlich weniger Überschneidungen mit roten Punkten aus dem Bereich 01c04\_04. Die Werte diesen Bereichs finden sich eher an den Rändern der blauen Gruppe. Auch die cyanfarbene Gruppierung ist im Vergleich zu vorher weniger stark mit roten Punkten durchsetzt. Insgesamt ist daher eine bessere Unterscheidung der einzelnen Gruppierungen zu erwarten.

Die Tabellen 4.5 und 4.6 spiegeln dies im Testergebnis wider. Die deutliche Steigerung der Konfidenz bei der Verwendung eines einfachen Medians ist in dieser Form jedoch nicht zu erwarten gewesen. Im Vergleich zum Mittelwert wurde eine Steigerung um 0,17 auf eine Konfidenz von 0,77 erreicht. Gleichzeitig konnte der Abstand der beiden Bereiche mit der höchsten Konfidenz auf 0,69 erhöht werden. Es lässt sich daraus folgern, dass die Zuordnung des richtigen Bereichs deutlich sicherer ist als mit dem Mittelwert.

Ebenfalls konnte die Konfidenz mit einem gleitenden Fenster verbessert werden. Auch hier erzielt der Einsatz des gleitenden Medians ein besseres Ergebnis als zuvor, kann jedoch nicht mit dem Ergebnis des einfachen Medians mithalten. Dies lässt darauf schließen, dass sich ein gleitender Median schlechter auf die Positionsbestimmung auswirkt als ein gleitender Mittelwert, da er im Vergleich zum einfachen Median deutlich stärker abfällt als die entsprechenden Mittelwerte.

Genau das gleiche Vorgehen ist auf die Quantile der Signal-Captures anwendbar. Auch sie scheinen in Abbildung 4.7 charakteristisch für die Verteilung zu sein. Daher werden sie ebenfalls näher betrachtet. Abbildung 4.9 zeigt einen Ausschnitt aus dem mit 25%-Quantilen generierten Merkmal-Raum.

Dieser weist große Ähnlichkeiten zum Median-Merkmal-Raum auf. Allerdings überschneiden sich die einzelnen Bereiche teilweise stärker als zuvor. Die zuvor gewonnene bessere Trennung von blauen und roten Bereichen ist nicht mehr vorhanden. Dafür ist die Trennung von cyan und rot deutlich besser, sodass es kaum noch zu Überschneidungen

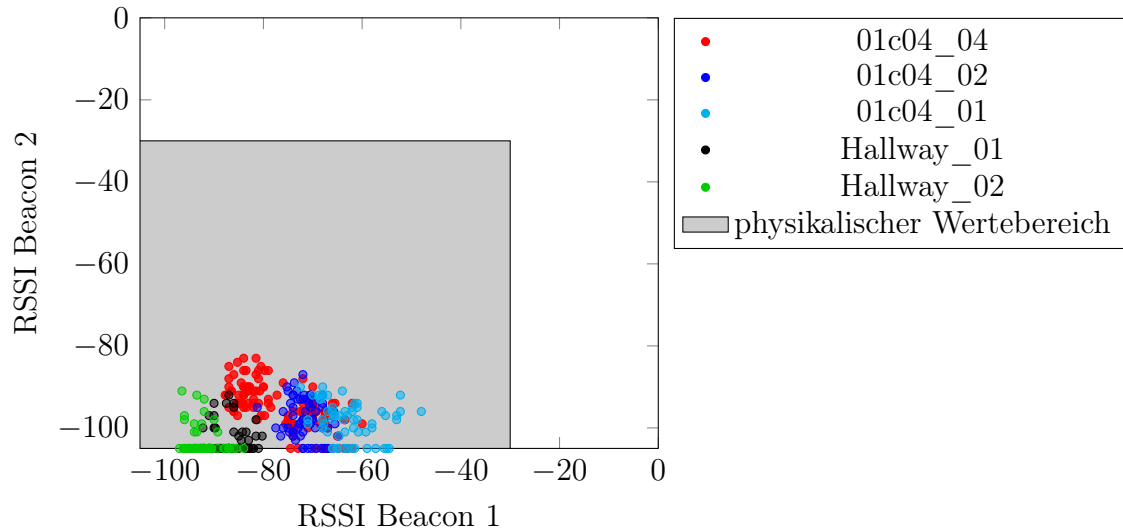


Abbildung 4.9: Die Abbildung zeigt den Merkmalraum, der durch das 25%-Quantil der Signal-Captures entsteht. Die roten und blauen Bereiche überdecken sich stärker als zuvor beim Median. Dennoch sind einzelne Gruppierungen enger zusammengedrückt.

kommt.

Im Integrationstest ist daher zu erwarten, dass es zu schlechteren Ergebnissen kommt als mit dem Median, da die Bereiche 01c04\_2 und 01c04\_4 schlechter auseinander gehalten werden können. Würde die Signal-Capture, gegen die getestet wird, aus dem Bereich 01c04\_01 stammen, würden bessere Ergebnisse prognostiziert werden können. Dennoch sollte der Bereich 01c04\_01 eine geringe Konfidenz erhalten, da sich dieser Bereich sehr gut von dem blauen Bereich 01c04\_02 trennen lässt.

Die Ergebnisse der Bereichszuordnung sind in den Tabellen 4.7 und 4.8 dargestellt. Der Bereich 01c04\_02 wird von in beiden Fällen korrekt als Bereich mit der höchsten Konfidenz wiedergegeben. Die Konfidenz besitzt insgesamt ein ähnliches Niveau wie beim Mittelwert. Allerdings ist die Differenz zwischen den beiden besten Bereichen etwas geringer. Ohne gleitendes Fenster liegt sie bei 0,28 und erfüllt somit das zweite Kriterium der Anforderung 4.1.1 nicht mehr.

Dieser Ansatz ist daher schlechter zu bewerten als die vorherigen Ansätze. Dennoch hat sich in Abbildung 4.9 gezeigt, dass die Verwendung des 25%-Quantils für bestimmte Bereiche funktionieren kann. Daher wird dieser Ansatz zur Merkmal-Vektor-Erstellung nicht vollständig fallen gelassen, sondern nach dem Praxistest und einem Feldversuch neu bewertet und mit den anderen Verfahren verglichen. Des Weiteren kann erwogen werden, ein Verfahren zu entwickeln, das mehr als nur eine Eigenschaft einer Verteilung verwendet. In einem solchen Verfahren kann auch das 25%-Quantil zu einem verbesserten Ergebnis beitragen, obwohl es alleine nur zu mäßigem Erfolg führt.

Als letztes statistisches Merkmal zur Merkmal-Vektor-Erstellung soll das 75%-Quantil untersucht werden. In Abbildung 4.10 ist ein Ausschnitt des dadurch entstandenen

Bereich	Konfidenz
01c04_02	0,58
01c04_04	0,30
01c04_01	0,03
01c04_03	0,03
01c04_05	0,02
01c04_06	0,01
...	

Tabelle 4.7: Testergebnisse mit einfachem 25%-Quantil

Bereich	Konfidenz
01c04_02	0,64
01c04_04	0,28
01c04_01	0,06
01c04_06	0,01
01c04_05	0,01
01c04_03	0,00
...	

Tabelle 4.8: Testergebnisse mit gleitendem 25%-Quantil

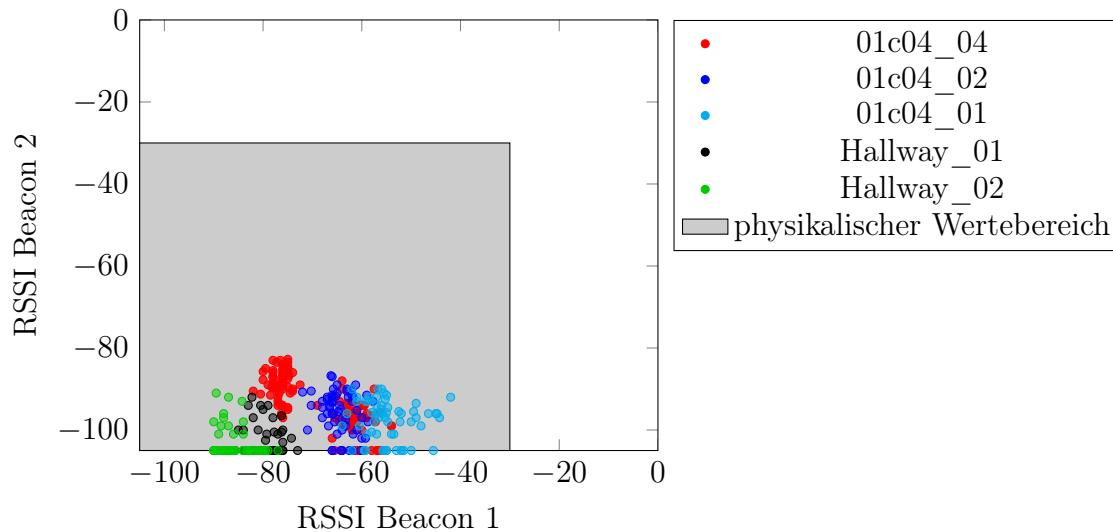


Abbildung 4.10: Dargestellt ist der Merkmalraum auf Basis des 75%-Quantils. Deutlich zu sehen ist, dass die roten und blauen Punkte dicht zusammengedrückt sind und die roten Punkte sich in zwei Häufungen aufteilen.

Merkmal-Vektor-Raumes zu sehen. Dabei ist auffällig, dass die einzelnen Punkte einer Farbe viel enger beieinander liegen, als sie es zuvor taten. Auch besitzen die roten, grünen und schwarzen Punkte auf der linken Seite ( $RSSI < -70$ ) des Bildes fast keine Überschneidungen. Auf der rechten Seite hingegen sind noch immer deutliche Überlappungen der Bereiche vorhanden.

Dennoch sind die Häufungen enger zusammengedrückt als beispielsweise beim Median. Daraus können sich vor allem im Hinblick auf die Unterscheidung von den Bereichen 01c04\_4 und 01c04\_2 Probleme ergeben, da der blaue Bereich große Überschneidungen mit dem roten Bereich besitzt. Ebenfalls zum Problem kann das dichtere Heranrücken der cyanfarbenen Punkte an die rotblaue Gruppe werden, da sie sich zwischen ihnen schlechter differenzieren lässt.

Aus den Ergebnissen des Integrationstests ergeben sich die Tabellen 4.9 und 4.10. In beiden Fällen erreicht der Bereich 01c04\_02 die höchste Konfidenz. Ohne gleitendes

Bereich	Konfidenz
01c04_02	0,66
01c04_04	0,19
01c04_01	0,07
01c04_03	0,04
01c04_05	0,02
01c04_06	0,01
...	

Tabelle 4.9: Testergebnisse mit einfachem 75%-Quantil

Bereich	Konfidenz
01c04_02	0,54
01c04_01	0,24
01c04_05	0,10
01c04_04	0,05
01c04_06	0,02
01c03_03	0,02
...	

Tabelle 4.10: Testergebnisse mit gleitendem 75%-Quantil

Fenster ist das Ergebnis zwischen Median und Mittelwert einzuordnen. Die Konfidenz von 0,66 ist schlechter als beim Median und besser als beim Mittelwert.

Gleiches gilt für den Abstand zum Bereich mit der zweithöchsten Konfidenz. Die Klassifikation über das einfache 75%-Quantil kann daher als Verbesserung gegenüber dem arithmetischen Mittel gewertet werden, auch wenn keine Verbesserung gegenüber dem Median vorliegt.

In Tabelle 4.10 sind deutlich schlechtere Ergebnisse zu sehen. Es konnte nur eine Konfidenz von 0,54 für den Bereich 01c04\_02 erzielt werden, was nach der Modelloptimierung die geringste, aufgetretene Konfidenz ist. Trotzdem wird die Anforderung 4.1.1 erfüllt. Daher ist auch die Merkmal-Vektor-Erstellung mit dem 75%-Quantil positiv zu bewerten, auch wenn Schwächen bei der Klassifikation mit gleitendem Fenster vorhanden sind. Ebenfalls hervorzuheben ist, dass im Integrationstest alle überprüften Ansätze den korrekten Raum nach der Modelloptimierung mit einer Wahrscheinlichkeit von 98%-99% finden. Dies liegt über den bisher möglichen 95%.

Auch bei diesem Ansatz deutet sich an, dass es zu genaueren Ergebnissen führen kann, wenn eine Kombination mehrerer statistischer Eigenschaften verwendet wird, um ein Signal einem Bereich zuzuordnen. Das heißt, die Merkmal-Vektoren der getesteten Verfahren (Median und Quantile) werden in einen gemeinsamen Merkmal-Vektor umgewandelt. Dadurch können alle drei Eigenschaften gleichzeitig in die Positionsbestimmung einbezogen werden. Da selbst ein Ausschnitt aus dem dadurch entstehenden Raum zu unübersichtlich ist, wird auf eine grafische Darstellung verzichtet.

Aus den Ergebnissen in den Tabellen 4.11 und 4.12 geht hervor, dass die Erstellung ohne gleitendes Fenster minimal schlechter ist, als die Verwendung eines Medians. Mit gleitendem Fenster ist sie als minimal besser zu bewerten, da der Abstand zwischen den beiden besten Bereichen um 0,08 größer ist als beim gleitenden Median.

Die Verwendung einer Kombination von arithmetischem Mittel und Median führt zu beinahe identischen Ergebnissen, weshalb diese nicht dargestellt werden. Anhand dieses Integrationstests lässt sich daher nicht sagen, ob ein kombiniertes Erstellen des Merkmal-Vektors sinnvoll ist. Dies muss sich über Praxistests und Feldversuche herausstellen. Bis hierhin erscheint der Median jedoch als prägendes Merkmal.

Bereich	Konfidenz
01c04_02	0,74
01c04_04	0,10
01c04_01	0,07
01c04_03	0,03
01c04_05	0,02
01c04_06	0,02
...	

Tabelle 4.11: Ergebnisse mit kombinierten Merkmalen

Bereich	Konfidenz
01c04_02	0,66
01c04_01	0,15
01c04_04	0,07
01c04_05	0,03
01c04_03	0,02
01c03_06	0,02
...	

Tabelle 4.12: Ergebnisse mit gleitenden kombinierten Merkmalen

#### 4.4.2 Praxis und Integrationstest für bewegliche Sender

Da durch den vorherigen Test nur Anhaltspunkte zur Verbesserung der Positionsbestimmung gefunden werden konnten, werden die einzelnen Möglichkeiten im Folgenden in einem praxisnäheren Test evaluiert. Als Maßstab dient der Test aus Abschnitt 4.1.3. Dabei wird die relative Häufigkeit richtiger Bereichszuordnungen bewertet. Die vorherigen Tests liefern zunächst einen Anhaltspunkt zu den Erwartungen.

Da allerdings ein solcher Test bereits mit dem Mittelwert durchgeführt wurde, lassen sich Vergleiche zu diesem ziehen. Werden alle Ergebnisse der Integrationstests mit dem des Mittelwertes verglichen, können Erwartungen an den Praxistest formuliert werden. Für diesen Praxistest werden keine gleitenden Fenster berücksichtigt, da sie im Integrationstest zumeist deutlich schlechter abschnitten als die Variante ohne gleitendes Fenster.

Wie in den vorherigen Tests wird mit dem Median begonnen. Dieser lieferte im Integrationstest mit Abstand die besten Ergebnisse. Daher sind auch für den Praxistest deutlich bessere Ergebnisse als im Integrationstest zu erwarten. Abbildung 4.11 zeigt die Ergebnisse für die einzelnen Bereiche im Praxistest. Werden sie direkt mit dem Mittelwert verglichen, zeigt sich, dass die Ergebnisse in der Praxis sichtbar schlechter sind und vor allem das gleiche Problem mit dem Bereich 01c04\_5 besteht. So liefern die Eckbereiche 01c04\_3 und 01c04\_6 annähernd die gleichen relativen Häufigkeiten wie beim Mittelwert.

Im Fall des Bereichs 01c04\_2 sinkt die Häufigkeit der richtigen Zuordnung um 2%. Gleichzeitig sinkt die Häufigkeit falscher Zuordnungen zu dem Bereich 01c04\_3 in Abbildung 4.11c um 9%. Die zuvor diesem Bereich falsch zugeordneten Signal-Captures verteilen sich allerdings auf alle anderen falschen Bereiche, sodass im Endeffekt zu erwarten ist, dass das im Feldversuch beobachtbare Flackern in einem größeren Radius auftreten wird.

Gleiches gilt für die Testsignale aus dem Bereich 01c04\_5. Positiv an der zugehörigen Abbildung 4.11c zu bewerten ist der geringere Anteil an fälschlicherweise dem Bereich 01c04\_2 zugeordneten Signal-Captures. Nichtsdestotrotz besitzt dieser Anteil mit 29% die größte Häufigkeit. Die im Vergleich zum Mittelwert fehlenden Anteile sind aller-

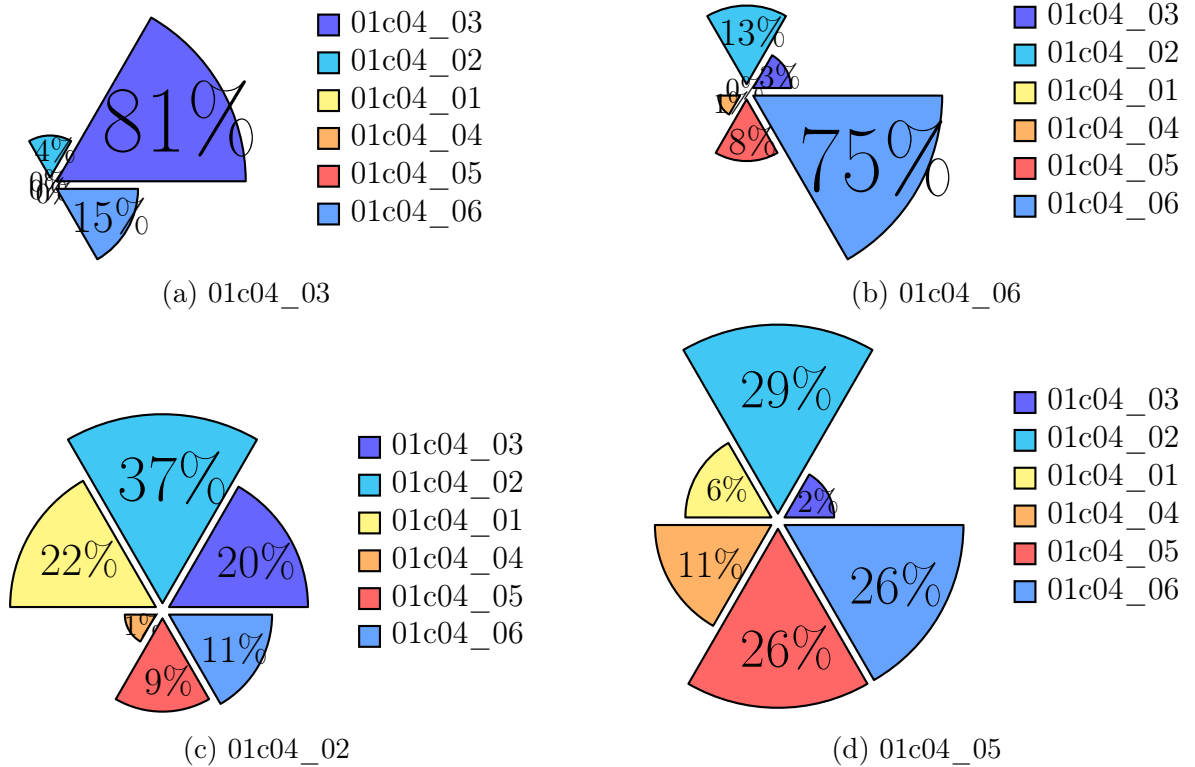


Abbildung 4.11: Abgebildet sind die relativen Häufigkeiten der ermittelten Bereichszuordnungen aus dem Praxistest über den Median als genutztes Transformationsmerkmal. Den Bildunterschriften ist zu entnehmen, aus welchem Bereich die zu klassifizierenden Signal-Captures des Positionsdatsatzes stammen. Das Ergebnis ist besser je höher der Anteil der Zuordnungen zum gleichnamigem Bereich der Bildunterschrift ist. Die einzelnen Tortenstücke sind analog zu den Raumbereichen angeordnet.

dings nicht richtig klassifiziert worden, sondern finden sich in den Anteilen der anderen falsch zugeordneten Bereichen wieder, was in der Realanwendung ein größeres Flackern bedeutet.

Im Feldversuch ließ sich durch die Verwendung des Medians anstelle des Mittelwertes keine spürbare Verbesserung erzielen. Das subjektiv empfundene Flackern war in beiden Fällen in einem ähnlichen Maße vorhanden. Während es in den Randbereichen wie zuvor als nur geringfügig empfunden wurde, konnten die Bereiche in der Raummitte nur selten richtig erkannt werden. Aus diesem Grund ist der Median nicht als Verbesserung der Ortungsleistung zu sehen.

Erwähnt werden sollte jedoch, dass der Median auch im Praxistest immer eine höhere Konfidenz für den wahrscheinlichsten Bereich lieferte, als der Mittelwert. In den meisten Fällen lag sie zwischen 0,7 und 0,95. Bei falsch zugeordneten Gebieten, lag sie dagegen meistens bei unter 0,6, wobei das richtige Gebiet häufig die zweithöchste Konfidenz besaß. Die Differenz der beiden Ergebnisse lag dabei um ca. 0,3. Dies lässt Raum



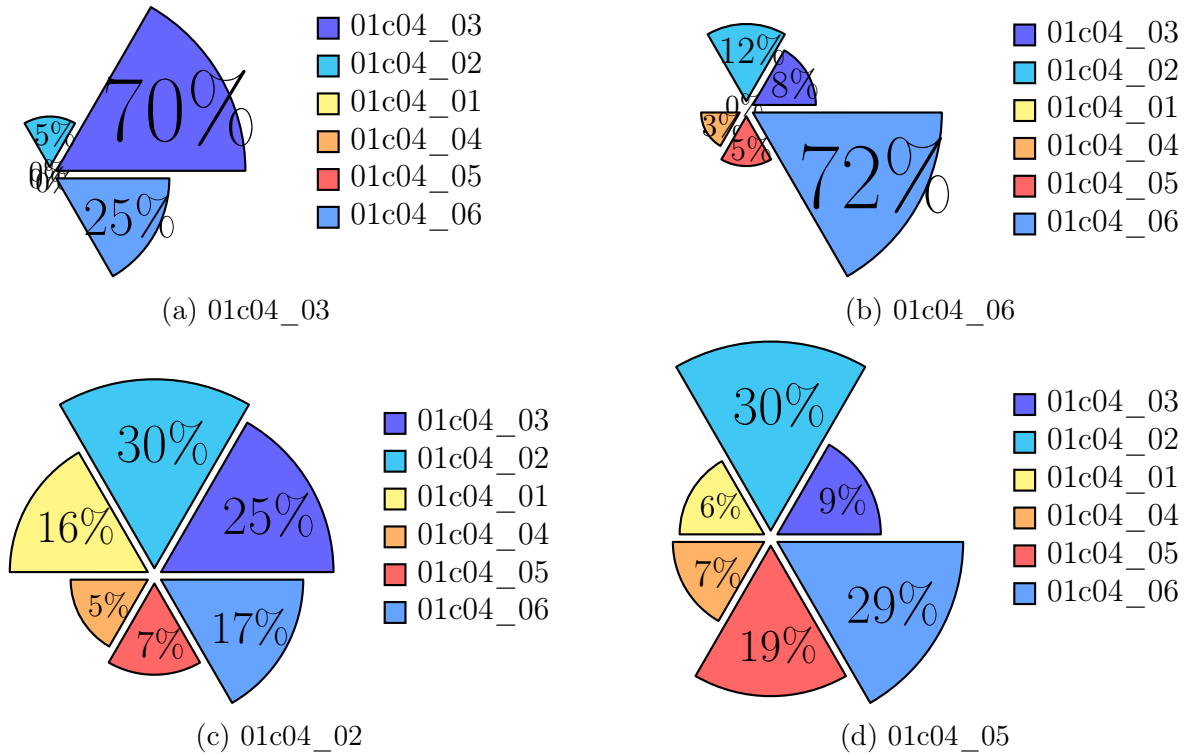


Abbildung 4.12: Dargestellt sind die relativen Häufigkeiten der ermittelten Bereichszuordnungen aus dem Praxistest über das 25% Quantil als genutztes Transformationsmerkmal

für heuristische Verbesserungsmöglichkeiten, die in dieser Ausarbeitung allerdings nicht weiter verfolgt werden.

Die Ergebnisse der 25%- und 75%-Quantile unterscheiden sich deutlich von den bisherigen. Vor allem das 25%-Quantil schneidet in Relation zu den bisherigen Resultaten weitaus schlechter ab (vgl. Abbildung 4.12). Zwar werden auch hier Eckbereiche (Abbildung 4.12a und 4.12b) vergleichsweise gut erkannt, allerdings sinkt die Häufigkeit der Erkennung auf knapp 70%.

Dies bedeutet bereits in gut erkennbaren Bereichen einen Genauigkeitsverlust von über 10%. Einen ähnlichen Genauigkeitsverlust erleidet ebenfalls die korrekte Zuordnung der mittleren Bereiche 01c04\_02 und 01c04\_05. Der Bereich 01c04\_05 wird, wie bereits zuvor, sehr schlecht erkannt. Allerdings können mit dem 25%-Quantil weniger als 19% der Testdaten dem korrekten Bereich zugeordnet werden (vgl. Abbildung 4.12d).

Für den Feldversuch ist daher in diesem Bereich nicht damit zu rechnen, dass eine ansatzweise korrekte Positionsbestimmung erfolgen kann. Das Flackern wird deutlich über einer richtigen Ortung liegen. Entsprechendes gilt auch für den Nachbarbereich 01c04\_02, dessen Signal-Captures zwar mit einer Häufigkeit von 30% richtig zugeordnet werden konnten (Abbildung 4.12c), jedoch verteilen sich die restlichen falschen Klassifikationen über den gesamten Raumbereich.

Dadurch ist in beiden mittleren Bereichen ein starkes Flackern über den gesamten

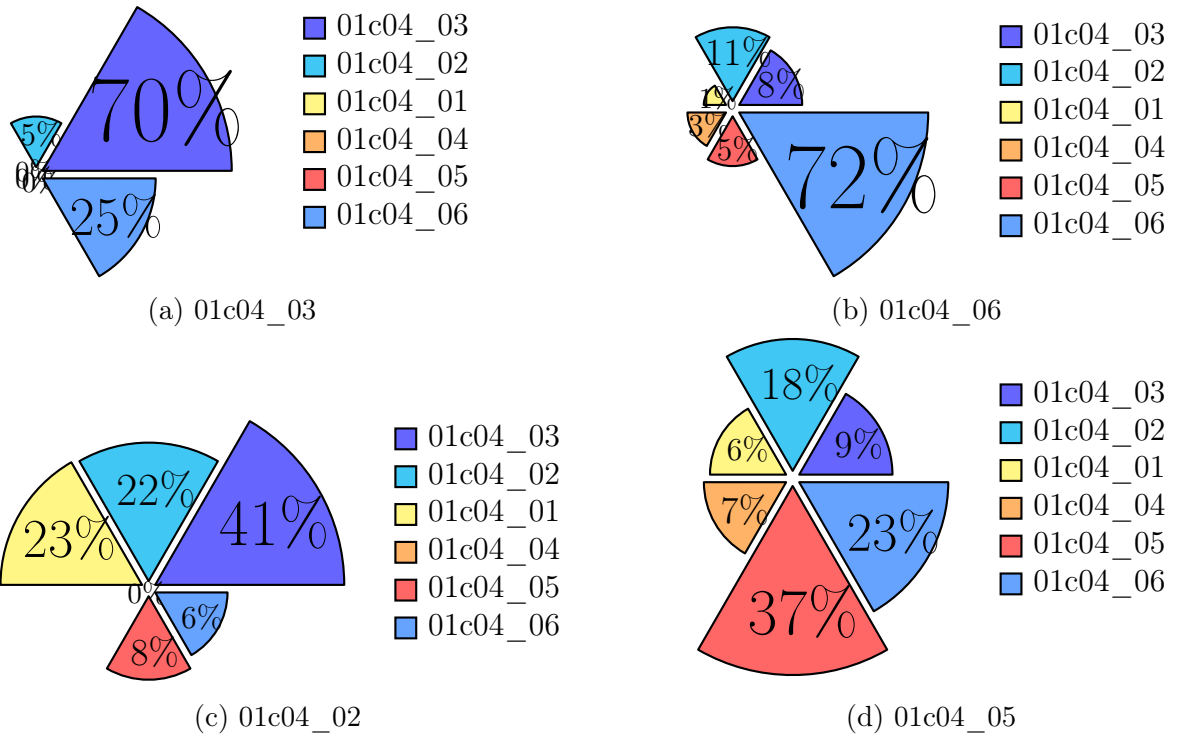


Abbildung 4.13: Abgebildet sind die relativen Häufigkeiten der ermittelten Bereichszuordnungen aus dem Praxistest über das 75% Quantil als genutztes Transformationsmerkmal

Raum zu erwarten. Das bedeutet, dass eine genaue Positionsbestimmung im Feldversuch nicht möglich ist, insbesondere nicht in der Raummitte. Dort scheint die Klassifikation der Daten teilweise zufällig zu erfolgen, da einfaches Raten den richtigen Bereich mit einer Wahrscheinlichkeit von 16,6% trifft. Deshalb ist der Ansatz über das 25%-Quantil als schlecht zu bewerten, was sich allerdings schon im Integrationstest angedeutet hatte. Im Feldversuch wurden diese Annahmen vollständig bestätigt. Das 25%-Quantil kann daher nicht eigenständig zur Klassifikation eingesetzt werden.

Wird das 75%-Quantil dem Praxistest unterzogen, ergeben sich die Ergebnisse aus Abbildung 4.13. Zunächst sind für die Eckbereiche in den Abbildungen 4.13a und 4.13b die gleichen Ergebnisse zu beobachten wie mit dem 25%-Quantil. Ein deutlicher Unterschied ergibt sich jedoch in den Lokalisierungsergebnissen der mittleren Bereiche. Zum ersten mal können Testdatensätze aus dem Bereich 01c04\_5 diesem auch mit 37% am häufigsten zugeordnet werden (vg. Abbildung 4.13d).

Zwar können diese 37% noch nicht als Erfolg gewertet werden, dennoch liefern diese Ergebnisse ein Anzeichen dafür, dass eine genauere Lokalisierung in den mittleren Bereichen möglich ist. Den Gegensatz zu diesem Ergebnis liefert Abbildung 4.13c, die zeigt, dass eine zuvor noch vergleichsweise gute Zuordnung im Bereich 01c04\_2 über das 75%-Quantil nicht mehr erreichbar ist.

Die Abbildung verdeutlicht, dass die Probleme, die zuvor bei der Klassifikation von

Signal-Captures aus dem Bereich 01c04\_5 in den Bereich 01c04\_2 verschoben wurden, da nun zu 41% der Bereich 01c04\_3 ermittelt wird, während auf den richtigen Bereich 01c04\_2 nur 22% der Testsignale kommen. Deshalb ist auch hier damit zu rechnen, dass in den mittleren Bereichen eine Ortung nur unzureichend funktioniert und es zu einem starken Flackern über den Raumbereich kommt. Auch dies wurde durch den Feldversuch bestätigt.

Aufgrund dieser Ergebnisse ist davon auszugehen, dass die beiden Quantile nicht alleine zur Ortung geeignet sind. Die Vermutung aus den Integrationstests wurde damit bestätigt. Allerdings konnte nicht bestätigt werden, dass die Ortung über den Median tatsächlich besser als über Mittelwert funktioniert, wie der Integrationstest zunächst vermuten ließ.

Des Weiteren ist die kombinierte Merkmal-Vektor-Erstellung aus 25%-Quantil, Median und 75%-Quantil dem Praxistest zu unterziehen. Die Ergebnisse aus den Zuordnungen sind in Abbildung 4.14 zu finden. Dabei weisen die Eckbereiche in den Abbildungen 4.14a und 4.14b wie beim Median und Mittelwert sehr gute Werte auf.

Bemerkenswert ist, dass der Median alleine im Bereich 01c04\_03 nur auf eine Genauigkeit von 81% besitzt, während das kombinierte Verfahren auf 83% kommt, was auf dem Niveau des Mittelwertes liegt. Im Bereich 01c04\_06 tritt dagegen keine nennenswerte Verbesserung auf. Dies bedeutet, dass trotz der Einbeziehung der Quantile die Klassifikation von Eckbereichen nicht zu einem schlechteren Ergebnis führt.

Bei Betrachtung der Ergebnisse für den Bereich 01c04\_05 (vgl. Abbildung 4.14d) wiederholen sich bisher bekannte Ergebnisse. Die Testdaten aus diesem Bereich werden hauptsächlich auf drei Bereiche verteilt: 01c04\_02, 01c04\_05 und 01c04\_06. Die meisten Daten werden mit dem kombinierten Verfahren mit 29% tatsächlich dem Bereich 01c04\_05 zugeordnet.

Dies stellt vor allem im Vergleich zum Mittelwert und Median eine Verbesserung dar, da Daten aus diesem Bereich zuvor nur mit 27% bzw. 26% zugeordnet wurden bei der gleichzeitigen Falschzuordnung von 36% bzw. 29% der Daten zum Bereich 01c04\_02. Dies ist auf den Einfluss des 75%-Quantils zurückzuführen, das 37% der Daten dem Bereich 01c04\_05 zuordnen konnte. Dennoch ist in diesem Bereich mit einem deutlichen, für den Menschen nicht zu unterscheidendem, Flackern während einer Live-Ortung zu rechnen.

Im Bereich 01c04\_02 konnten 5% (Mittelwert) bzw. 3% (Median) der Testdaten weniger korrekt zugeordnet werden. Auch hier ist der Einfluss des 75%-Quantils gut erkennbar. Dieser ordnete 41% der Signale dem falschen Bereich 01c04\_03 zu. Dies führt im kombinierten Verfahren dazu, dass der Anteil dieser Zuordnungen von 20% beim Median auf 25% im kombinierten Verfahren steigt. Dass dieser Einfluss nicht zu schlechteren Ergebnissen führt, liegt am 25%-Quantil, das ebenfalls ein Viertel der Testdaten diesem Bereich falsch zugeordnet hat. Aufgrund der Ergebnisse ist die Ortung im Bereich 01c04\_02 minimal schlechter als beim Mittelwert oder Median.

Der Feldversuch zeigte, dass die Eckbereiche – im Gegensatz zu den Verfahren mit einzelnen Quantilen – im kombinierten Verfahren die bereits bekannten Genauigkeiten und ein sehr geringes Flackern aufweisen. In den mittleren Raumbereichen ist eine Veränderung im Vergleich zum Mittelwert und Median ebenfalls subjektiv nicht spürbar,

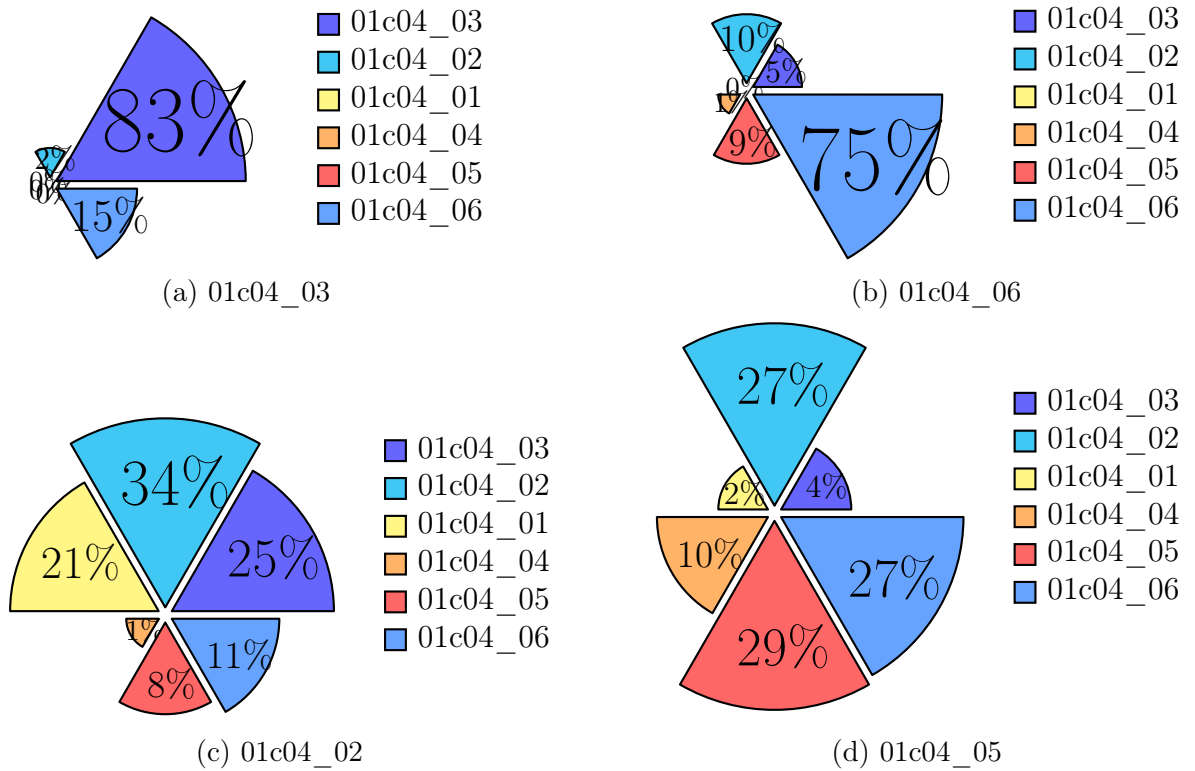


Abbildung 4.14: Die Abbildung zeigt die Ergebnisse des Praxistests, die durch die Kombination mehrerer Merkmale (Median, 25%- und 75%-Quantil) gewonnen werden. Im Vergleich zu den anderen Ergebnissen führen die Stärken der einzelnen Verfahren zum bisher besten Ortungsergebnis, während die Schwächen nur gering auf die Klassifikationsleistung einwirken.

auch wenn sie eindeutig messbar ist. Zwei bis fünf Prozentpunkte mehr oder weniger korrekte Zuordnungen im Praxistest scheinen durch die zu geringe Grundgenauigkeit in diesen Bereichen einem Menschen bei einer Live-Lokalisierung nicht aufzufallen.

Erst ab einem Bereich um 70% erscheinen auch geringe Verbesserungen spürbar, da die Verbesserung zu den einzelnen Quantil-Verfahren bereits fühlbar ist. Dass die Ergebnisse sich dennoch im Feldversuch nicht merklich unterschieden haben, liegt daran, dass ähnliche Eigenschaften zur Merkmal-Vektor-Erstellung genutzt wurden.

Dadurch sind selbst im kombinierten Verfahren die Unterschiede in den Vektorräumen zu gering, um vor allem in schlecht trennbaren Bereichen zu einer guten Klassifikation zu führen. Auf Basis der vorgestellten Ergebnisse ist abzuleiten, dass die Kombination mehrerer Eigenschaften zu einer verbesserten Positionsbestimmung führen kann.

### 4.4.3 Mehrheitsvotum

Anstelle eines einzelnen Klassifikators, der mehrere Merkmale bei der Transformation der Signal-Captures in den Merkmalraum berücksichtigt, können auch mehrere Klassifikatoren genutzt werden. Diese werden als Gruppe zusammengefasst und bilden einen neuen Klassifikator. Das einfachste Verfahren der sog. Ensemble-Techniken ist das Mehrheitsvotum.

Dabei wird eine Signal-Capture durch jeden Klassifikator einer Gruppe einem bestimmten Bereich zugeordnet. Anschließend kann geprüft werden, welcher Bereich von der Klassifikatorgruppe am häufigsten gewählt wurde. Dieser wird als Ergebnis des Gesamt-Klassifikators ausgewählt.

Die Annahme hinter einem solchen Verfahren ist, dass eine solche Gruppe ein genaueres Ergebnis liefert als ein einzelner Klassifikator. Dieses kann allerdings nur verbessert werden, wenn die einzelnen Klassifikatoren einander widersprechen. Würden zum Beispiel alle Klassifikatoren die gleiche (falsche) Entscheidung treffen, wäre ein Mehrheitsvotum nicht nützlich.

Wird ein Mehrheitsvotum jedoch auf Basis von unkorrelierten und gleichzeitig unabhängigen Klassifikatoren durchgeführt, dürfen einzelne Klassifikatoren mit ihrer Entscheidung falsch liegen, solange die Mehrheit diese Entscheidung korrigiert. Dadurch kann die Fehlerrate des Gesamtverfahrens deutlich unter der Fehlerrate der einzelnen Klassifikatoren liegen.

Ein naiver erster Versuch wäre die erneute Kombination von Median, 25%- und 75%-Quantil. Bei einem Mehrheitsvotum werden die Merkmale im Gegensatz zur vorherigen Verfahren jedoch nicht dazu genutzt, einen gemeinsamen Merkmalraum zu erstellen, sondern zu jedem Merkmal wird ein eigenständiger, unabhängiger Merkmalraum gebildet. Daraus resultiert eine Support Vector Machine pro Merkmalraum. Wichtig ist darauf zu achten, dass alle SVMs mit demselben Datensatz trainiert werden. Werden jedoch genau diese drei Merkmale betrachtet, lässt sich sagen, dass sie zu einem gewissen Grad korreliert sind.

Abbildung 4.15 zeigt die Ergebnisse des Praxistests für ein Mehrheitsvotum mit den drei genannten Merkmalen. Werden die Ergebnisse mit denen aus Abbildung 4.14 verglichen, fällt auf, dass die Ergebnisse aus den Eckbereichen nur um einen Prozentpunkt schlechter sind. In Abbildung 4.15d zeigt sich allerdings, dass dieser Bereich mit einem Mehrheitsvotum um 12% besser erkannt wurde und zum ersten mal ein Bereich aus der Raummitte über 40% korrekt zugeordnete Signal-Captures erreichen konnte. Hier zeigt sich das Potenzial des Votums gegenüber einzelnen Klassifikatoren.

In Abbildung 4.15c ist jedoch zu sehen, dass ein Mehrheitsvotum nicht immer gleichbedeutend mit besseren Ergebnissen ist. Im Bereich 01c04\_02 fiel die Erkennungsrate von 34% auf 25%. Die meisten Signal-Captures aus diesem Bereich werden nun mit 26% dem Bereich 01c04\_03 zugeordnet. Durch die Ausprägung des Kreises ist in diesem Bereich in einem Feldversuch mit starken Flackern zu rechnen.

Ein Mehrheitsvotum, bei dem darauf geachtet wird, dass die einzelnen Klassifikatoren zusätzlich zur Unabhängigkeit auch unkorreliert sind, kann zu deutlich besseren Ergebnissen führen, als bisher durch einzelne Klassifikatoren möglich war. Die Ergebnisse eines

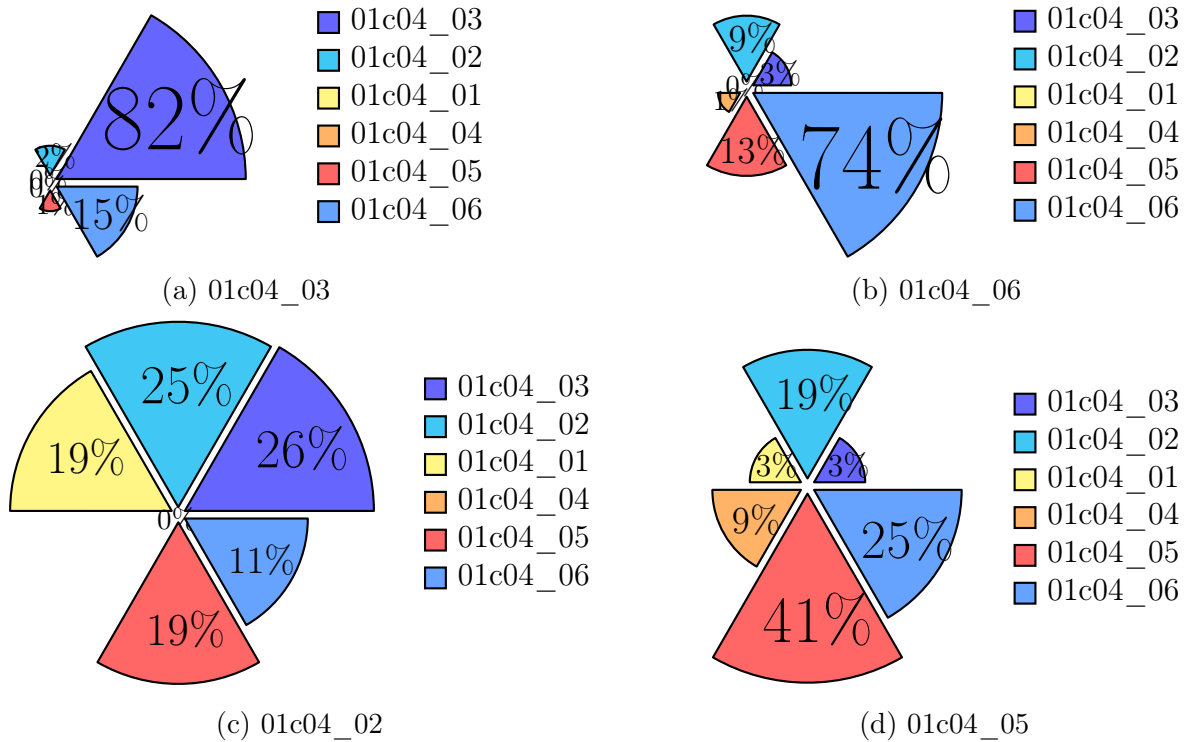


Abbildung 4.15: Die Abbildung zeigt die Ergebnisse des Praxistests, die durch die Kombination mehrerer Merkmale (Median, 25%- und 75%-Quantil) in einem Mehrheitsvotum erreicht werden. Durch die enge Korrelation der Merkmale ergeben sich deutlich schlechtere Zuordnungen als mit einem kombinierten Merkmalraum.

solchen Mehrheitsvotums sind in Abbildung 4.16 zu sehen.

Dazu wurden ebenfalls drei Klassifikatoren verwendet. Allerdings wurde neben den Quantilen auch der Mittelwert berücksichtigt. Ebenso wurden mit den bisher als schlecht eingestuft 25%- und 75%-Quantile ein kombinierter Klassifikator gebildet. Dennoch sind die Merkmale zunächst korreliert. Um die Korrelation der einzelnen Merkmale zu senken, wurde mit gleitenden Fenstern bei Median und Mittelwert gearbeitet.

Die Ergebnisse zeigen, dass bei einer geringen Korrelation der Merkmale, die positiven Aspekte des kombinierten Verfahrens noch einmal gesteigert werden konnten. Ebenso konnten die positiven Werte aus dem naiven Mehrheitsvotum konserviert werden, ohne dass die Ergebnisse anderer Bereiche verschlechtert wurden. Dadurch liegt die Erkennungsrate in den Eckbereichen jetzt über 80% und konnte im Bereich 01c04\_06 um 10% von 75% auf 85% gesteigert werden.

Die vorherigen Ergebnisse im Bereich 01c04\_05 konnten wiederholt werden. Damit wurde auch hier eine Verbesserung im Vergleich zur einfachen Merkmalkombination erreicht. Die Ergebnisse aus dem Bereich 01c04\_02 sind minimal besser als bei dem kombinierten Verfahren und deutlich besser als bei dem naiven Mehrheitsvotum.

Dadurch wurde gezeigt, dass ein Mehrheitsvotum mit möglichst unkorrelierten Klassi-

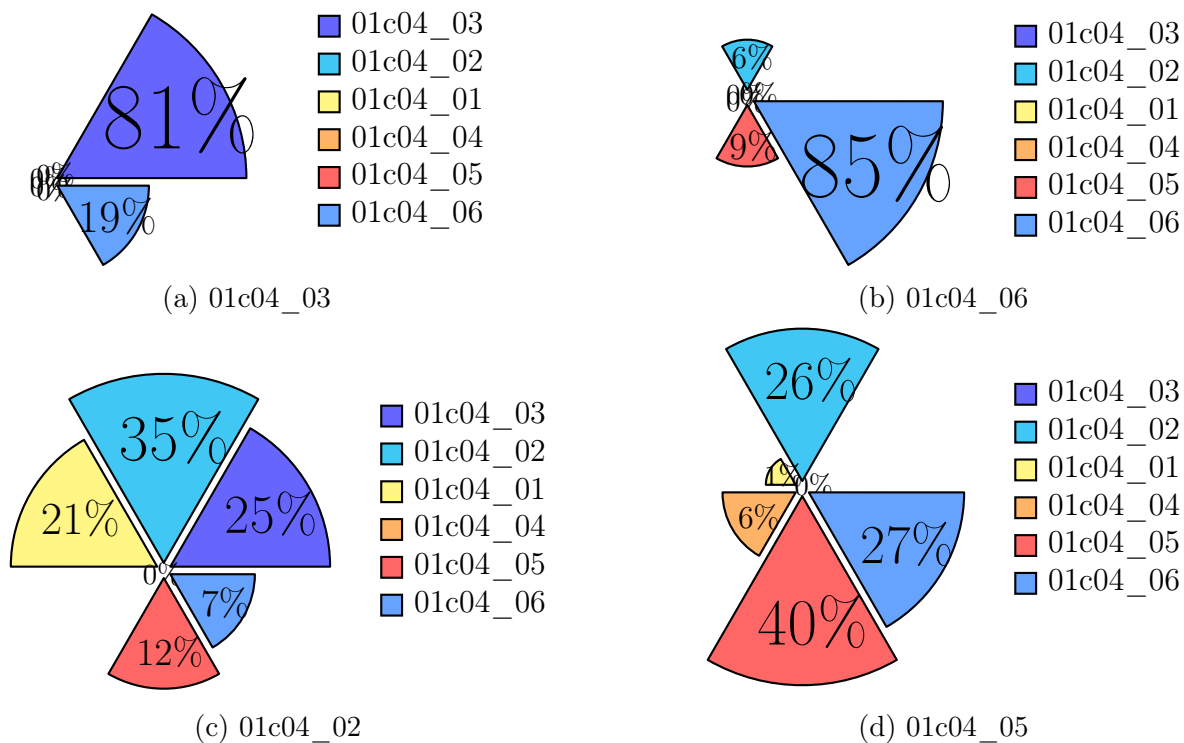


Abbildung 4.16: Die Abbildung zeigt die Ergebnisse des Praxistests, die durch die Kombination mehrerer Merkmale (Mittelwert, Median, 25%- und 75%-Quantil) in einem Mehrheitsvotum erreicht werden. Durch die Verwendung von verschiedenen gleitenden Fenstern wurde die Korrelation der Merkmale gesenkt, was zu erheblich besseren Ergebnissen führt.

fiktoren zu einer weiteren Verbesserung der Ortungsgenauigkeit beitragen kann. Insgesamt betrachtet konnte die Positionsbestimmung von einem undefinierten, nicht messbaren Flackern, bei dem der Ortungserfolg an Zufall erinnert, durch zahlreiche Optimierungen auf eine Genauigkeit von über 80% an den Randbereichen eines Raumes gesteigert werden.

In einem Feldversuch zeigte sich, dass die Ergebnisse sich durch ein Mehrheitsvotum nicht gravierend von den kombinierten Merkmalen unterscheiden. Insgesamt wirkt die Ortung durch die verwendeten gleitenden Fenster träger als zuvor. Bei einem längeren Aufenthalt in einem Bereich funktioniert die Lokalisierung rein subjektiv besser als zuvor.

## 4.5 Allgemeine Beobachtungen während der Experimente

Während der Feldversuche konnten neben den Ergebnissen des nachgestellten Praxistests weitere Szenarien innerhalb und außerhalb der Räume getestet werden. Die Ergebnisse werden in den folgenden Abschnitten dargestellt.

### 4.5.1 Beobachtungen innerhalb eines Raumes

Neben den messbaren Ergebnissen konnten während der Feldversuche viele weitere Eindrücke gewonnen werden. So ist die Position des Beacons während einer Lokalisierung entscheidend für eine genaue Ortung. Bei der Offline-Datenerhebung wurde das Beacon auf Kopfhöhe so positioniert, dass der eigene Körper das Signal möglichst wenig verfälscht. Wird während einer Ortung das Beacon an einer anderen Stelle positioniert bzw. in der Hand gehalten, wird das Flackern auf der Karte stärker. Es traten somit vermehrt falsche Bereichszuordnungen auf.

Außerdem konnte beobachtet werden, dass bei schnellen Bewegungen innerhalb eines Bereichs die Ortung kurzzeitig versagt. Die Signal-Capture, die aufgenommen wurde als die schnelle Bewegung stattfand, wurde meist nicht dem richtigen Bereich des Raumes zugeordnet. Dies war unabhängig von der Lage des Bereichs zu beobachten.

Weiterhin lies sich beobachten, dass je näher die zu verfolgende Person einem Raspberry Pi kam, die Positionsbestimmung sehr konstant dem richtigen Bereich zugeordnet wurde. Im Grenzbereich zwischen zwei oder mehr Gebieten nahm die Genauigkeit der Lokalisierung dagegen ab und das Flackern erhöhte sich vor allem zwischen den beiden Bereichen. Ausnahme bilden die mittleren Bereiche. Dort erstreckte sich das Flackern auch über Bereiche, die nicht direkt aneinander grenzen. Dies war jedoch in dieser Form zu erwarten.

Weitere Verfälschungen der Ortung konnten beobachtet werden, wenn andere Personen die Testumgebung betraten. Da in der Offline-Phase keine weiteren Personen anwesend waren, wurde der Einfluss solcher auf das Signal beim Training nicht berücksichtigt. Besonders stark trat dieser Effekt auf, wenn sich zusätzlich anwesende Personen in der Mitte des Raumes aufhielten. Je näher sie sich an eine Wand bewegten, ohne dabei ein Raspberry abzuschirmen, desto besser funktionierte die Ortung.

Während mit diesem Ergebnis gerechnet werden konnte, wenn ein Mensch sich in der direkten Sichtlinie zwischen Beacon und Raspberry aufhält, war jedoch überraschend, dass die Anwesenheit verschiedener Menschen ausreicht, um die Ortung eindeutig zu verschlechtern. Obwohl der Effekt bekannt war, wurde er zunächst als schwächer eingestuft.

Diese Erkenntnisse zeigen, wie wichtig die Offline-Phase für Fingerprinting-Verfahren ist und wie sensibel die Support Vector Machine auf bereits kleine Änderungen reagiert. Daher sollte immer darauf geachtet werden, die späteren Umstände bereits während der Offline-Phase abzubilden.

Während längerer Feldversuche konnte eine weitere Erkenntnis gewonnen werden: In



unregelmäßigen Abständen tritt eine fehlerhafte Kommunikation zwischen Server und den Raspberry Pis vereinzelt auf. Teilweise kann es vorkommen, dass ein Raspberry über mehrere Sekunden (bis zu 20) keine Nachricht über den RabbitMQ-Server an das Backend sendet, obwohl es theoretisch mindestens alle 700 ms die aufgenommenen Bluetooth-Signale verschicken sollte. Woran dies liegt, konnte aufgrund des unregelmäßigen Auftretens nicht abschließend geklärt werden.

Es ist jedoch zu vermuten, dass sich der Timer im Raspberry Pi gelegentlich „verschluckt“, da die Kapazität des RabbitMQ-Servers nicht ansatzweise ausgelastet wurde, der Client auf dem Raspberry nicht abstürzte, sondern irgendwann wieder die aufgenommenen Signale versendete, ohne dass ein Eingriff stattfand. Danach sendete es außerdem im vorgesehenen Zeitintervall, sodass die normale Ortung fortgesetzt werden konnte.

### 4.5.2 Beobachtungen bei Raumwechseln

Da im Rahmen dieser Ausarbeitung auch die Genauigkeit der Erkennung von Raumwechseln untersucht werden sollte, wurde zu diesem Anwendungsfall ebenfalls ein Feldversuch durchgeführt. Dazu wurde die Ortungsgenauigkeit bei einem Türübergang zwischen den Räumen 01c03 und dem Flur 01c sowie dem Übergang zwischen 01c04 und dem Flur über einen Zeitraum von mehr als einer Stunde untersucht.

Dabei wurde zunächst auf das klassische Szenario mit beweglichen Empfängern zurückgegriffen. Als Lokalisierer kam die SVM mit der Merkmalraumtransformation über Median, 25%- und 75%-Quantil ohne Mehrheitsvotum zum Einsatz. Innerhalb dieser Stunde wurde der Aufenthaltsbereich der Testperson nur in zwei Fällen über jeweils einem Capture-Intervall falsch bestimmt. Daher lässt sich sagen, dass Räume mit beweglichen Empfängern nahezu sicher erkannt werden können.

Ebenfalls wurde ein Feldversuch in kleinerem Rahmen mit beweglichen Sendern durchgeführt. Aufgrund der eingeschränkten Anzahl an Raspberry Pis und der fehlenden Möglichkeit, Strom in den Flur 01c zu verlegen ohne den Versuch zu verfälschen, wurde lediglich der Übergang von Raum 01c04 in den Bereich 01c\_06, der direkt vor der Tür des Raumes liegt, untersucht. Dabei war die Bestimmung des richtigen Raumes solange möglich, wie der Beacon auf dem Kopf der zu verfolgenden Person platziert war.

Sobald der Beacon in etwa auf Bauchhöhe platziert wurde, war eine genaue Zuordnung des Raumes nicht mehr möglich. Dies bestätigt die Ergebnisse aus dem vorherigem Abschnitt, dass die Position des Beacons maßgeblich zur Ortungsgenauigkeit beiträgt. Während des Feldversuchs wurde keine einzige Signal-Capture falsch zugeordnet, wenn das Beacon richtig positioniert wurde.

Ob dies der Genauigkeit des gewählten Verfahrens oder den dicken Wänden in der Testumgebung geschuldet ist, lässt sich nicht mit Sicherheit bestimmen. Sollten in einer anderen Umgebung Probleme mit der Erkennung von Raumwechseln oder im schlimmsten Fall ein Flackern über mehrere Räume auftreten, kann mit verschiedenen Ansätzen versucht werden, die Problematik zu reduzieren.

Für eine größere Sicherheit bei der Erkennung könnte ein Raumwechsel nur dann erlaubt werden, wenn mehrere Signal-Captures hintereinander dem gleichen Raum zugeordnet werden. Dadurch würde die Lokalisierung jedoch insgesamt träge. Für Anwen-

dungen, bei denen nur festgestellt werden soll, ob eine Person einen Raum betreten hat, wäre diese Lösung jedoch ausreichend.

In anderen Fällen könnte ermittelt werden, welcher Raum der Wahrscheinlichste ist, anstelle eines Gebietes. Nur wenn das Gebiet mit der höchsten Konfidenz zu dem Raum mit der höchsten Konfidenz passt, wird ein Wechsel des Raumes erlaubt. Der Raum mit der höchsten Konfidenz kann dazu über die Summe aller Teilgebiete eines Raumes bestimmt werden.

# 5 Zusammenfassung und Fazit

Im den folgenden Abschnitten werden die Ergebnisse und das gewählte Vorgehen abschließend zusammengefasst. Weiterhin werden bisher ermittelte Grenzen des entwickelten Systems dargestellt, wodurch sich Möglichkeiten und Einschränkungen für verschiedene Anwendungsfälle ergeben, die ebenfalls in diesem Kapitel behandelt werden.

## 5.1 Zusammenfassung

Im Verlauf dieser Ausarbeitung wurde das bisherige System zur Verfolgung von beweglichen Empfängern mit stationär installierten Bluetooth-Beacons der Fachhochschule Aachen um die Möglichkeit erweitert, dieses Szenario umzudrehen und somit bewegliche Beacons mit fest installierten Empfängern zu verfolgen. Dazu wurde zu Beginn der Arbeit auf grundlegende Eigenschaften des Bluetooth-Signals und den Aufbau von Bluetooth-Beacons eingegangen. Als Empfänger wurden Raspberry Pis mit Bluetooth- und WLAN-Adapter eingesetzt. Des Weiteren wurden mögliche Ursachen von Signalstörungen behandelt.

Danach wurde das vorhandene System zur Lokalisierung von Smartphones mit fest installierten Beacons, so erweitert, dass nun auch bewegliche Beacons mit fest installierten Empfängern durch eine SVM lokalisiert werden können. Dazu wurde erläutert, welche Erweiterungen der einzelnen Komponenten notwendig sind und wie die vorhandene Datenstruktur im *Location-Solver* weiterhin verwendet werden kann. Dadurch wurde ermöglicht, beide Anwendungen mit dem gleichen System zu betreiben, ohne dass eine komplette Neuimplementierung nötig wurde.

Ebenfalls wurde darauf eingegangen, wie die Signale in den verschiedenen Phasen des Fingerprinting synchronisiert werden können. Zudem wurde die eingesetzte Implementierung einem ersten Test unterzogen. Dabei wurde jedoch festgestellt, dass Optimierungen in der Ortungsgenauigkeit nötig sind, damit die Lokalisierung von beweglichen Beacons eine akzeptable Genauigkeit erreicht, da ansonsten lediglich ein nahezu zufälliges Flackern der Positionsmarkierung auf der Karte beobachtet werden konnte.

Aufgrund der zunächst schlechten Ortungsgenauigkeit wurde das System insbesondere auf die Optimierungsfähigkeit der Signal-Capture-Transformation in den Merkmalraum untersucht. Dazu wurde zunächst eine Testmethodik entwickelt, die auf drei einzelnen Tests besteht: Dem Integrationstest für bewegliche Empfänger, einem praxisnahen Test für stationäre Empfänger und einem Feldversuch, der keine objektiv messbaren Ergebnisse liefert, sondern subjektive Eindrücke des Ortungserfolges. Dadurch konnte gewährleistet werden, dass alle durchgeführten Analysen auf beide Szenarien gleichermaßen anwendbar sind. Untersucht wurde, welche Merkmale einer Signal-Capture zum besten

Ortungserfolg führen. Dabei stellte sich heraus, dass die Kombination mehrerer Eigenschaften stets einzelnen Eigenschaften überlegen ist.

Über ein Mehrheitsvotum konnte das Ergebnis aus der Lokalisierung über kombinierte Merkmale noch einmal gesteigert werden. Dabei ist jedoch darauf zu achten, dass Klassifikatoren eingesetzt werden, deren negativen Eigenschaften sich aufheben, sowie dass alle in etwa die gleiche Ortungsgenauigkeit besitzen. Nur in diesem Fall führt ein Mehrheitsvotum tatsächlich zu einer Verbesserung.

Ebenfalls konnte durch die einzelnen Feldversuchen festgestellt werden, dass bereits die Anwesenheit mehrerer Menschen genügt, um die Ortung zu beeinträchtigen, ohne dass sie in einer direkten Linie zwischen Sender und Empfänger stehen müssen. Eine Genauigkeit auf Raumebene ließ sich allerdings in allen Fällen gewährleisten, sodass die Ergebnisse dieser Arbeit besser sind als die bisher möglichen 95%. Ob dies allerdings an den dicken Wänden, die das Bluetooth-Signal stark abschirmen oder an der Genauigkeit des Verfahrens liegt, lässt sich nicht mit Sicherheit feststellen.

In dieser Arbeit konnte somit gezeigt werden, dass die Lokalisierung von Bluetooth-Beacons mit stationär installierten Empfängern durchaus möglich ist. Allerdings reagiert ein solcher Aufbau deutlich anfälliger auf verschiedene Einflüsse als das klassische Szenario.

### 5.2 Fazit: Grenzen des Systems

Einem Szenario zur Verfolgung von beweglichen Beacons sind allerdings vergleichsweise enge technische Grenzen gesetzt. Soll ein gesamtes Gebäude oder sogar ein Industriekomplex mit einem solchen Ortungssystem ausgestattet werden, so müssen vor allem sehr leistungsfähige Router eingesetzt werden, da jedes eingesetzte Raspberry eine eigene WLAN-Verbindung benötigt. Über diese werden sehr viele Daten ausgetauscht.

Darüber hinaus fiel bereits während der Feldversuche eine WLAN-Antenne aus. Dies könnte auf eine zu hohe Belastung hindeuten, obwohl die verschickte Datenmenge normalerweise kein Problem darstellen sollte. Weiterhin ist nicht bekannt, ab welcher Anzahl eingesetzter Raspberry Pis eine Überlastung des RabbitMQ-Servers auftritt bzw. die Verarbeitung im Backend zu langsam ist, um die Beacons innerhalb eines Capture-Intervall zu lokalisieren. Auch wie viele Beacons maximal gleichzeitig lokalisiert werden können, wurde nicht ausgereizt. Tests mit maximal zehn eingesetzten Beacons und vier Raspberry Pis funktionierten.

Eine weitere Limitierung ist die ständige Notwendigkeit einer stabilen Stromversorgung, da die Raspberry Pis keinen Akku besitzen. Dadurch sind sie ohne externe Stromversorgung auf ein Kabel angewiesen und nicht überall positionierbar, was bereits im Feldversuch für Raumwechsel auffiel.

Mit dem in dieser Arbeit entwickelten Ortungsverfahren kann der Aufenthalt einer zu verfolgenden Person auf Raumebene zuverlässig detektiert werden. Unterhalb dieser Ebene lassen sich mit dem Verfahren am besten einzelne Quadranten eines Raumes detektieren. Teilt man jeden Raum in maximal vier Abschnitte ein, ist davon auszugehen, dass ein entsprechender Abschnitt einer ungefähren Größe von  $3 \times 3$  Metern mit einer

Wahrscheinlichkeit von über 80% korrekt detektiert wird. Diese Einschränkung gilt jedoch für beide Szenarien und ist unabhängig davon, welcher Teil des Systems beweglich und welcher stationär ist.

## 5.3 Fazit: Mögliche Anwendungsgebiete

Durch die engen Grenzen werden auch die möglichen Anwendungsgebiete eingeschränkt. Dennoch ergeben sich durch die Ergebnisse verschiedenen Einsatzmöglichkeiten. Die erste, die bereits in der Motivation dieser Ausarbeitung genannt wurde, ist die experimentelle Verfolgung von Fußgängern.

Ein durchführbares Szenario sind Experimente zur Wegfindung von Personen, die sich dazu durch ein Gebäude bewegen. Dabei reicht die vorhandene Raumgenauigkeit aus, um Entscheidungen nachvollziehen zu können, da Raumwechsel zuverlässig erkannt werden können. Hierbei ist grundsätzlich der Einsatz von beweglichen Beacons als auch Smartphones als beweglichen Empfängern möglich. Für Experimente eignen sich Beacons allerdings eher, da sie die Personen nicht durch einen Bildschirm in ihrer Entscheidung beeinflussen können.

Ebenfalls interessant wäre jedoch auch den Einfluss von Hinweisen auf die Person zu ermitteln, die durch Schilder oder Informationen auf dem Smartphone entstehen. Dadurch wird eine kostengünstige Möglichkeit für experimentelle Aufbauten geboten, da der Aufbau im Vergleich zu sonst benötigten Kameras sehr preiswert ist. Weiterhin lassen sich solche Experimente auch in Dunkelheit oder in einer verräuchten Umgebung durchführen. Bei allen Versuchen ist jedoch darauf zu achten, dass die Beacons analog zur Offline-Phase richtig positioniert werden.

Des Weiteren könnten die Empfänger über den Köpfen der Personen an Decken angebracht werden, um Störungen zu minimieren. Außerdem können die Experimente nur bei einer geringen Personendichte durchgeführt werden. Im besten Fall werden Versuchspersonen einzeln durch das Gebäude geschickt, um die unerwünschte Beeinflussung der Signale durch andere anwesende Personen auszuschließen. Dies kann letztendlich zu einer besseren Modellierung des Routings von Fußgängern in einer Entfluchtungssimulation führen.

Die Ergebnisse zeigen allerdings auch, dass der Einsatz zur Navigation von Gabelstaplern in einem Warenlager mit der erreichten Genauigkeit nicht ratsam ist, da eine große Halle zu wenige Randbereiche besitzt, in denen die Lokalisierung zuverlässig funktionieren kann. Um ein solches Anwendungsszenario umzusetzen, müsste die Genauigkeit in Raumbereichen, die mittig im Raum liegen, ebenfalls auf ca. 80% erhöht werden.

Des Weiteren ist die größere Geschwindigkeit der Gabelstapler im Vergleich zu einem Menschen ebenfalls problematisch, da die Ortungsgenauigkeit mit zunehmender Geschwindigkeit geringer wird. Mit Hilfe weiterer Sensoren könnte die Positionsbestimmung wie in Abschnitt 6.2 beschrieben verbessert werden.



## 6 Ausblick

Abschließend sollen verschiedene Aspekte der Ausarbeitung herausgegriffen und Ansätze aufgegriffen werden, wie in der Zukunft Verbesserungen erreicht werden könnten. Dabei soll zum einen auf die erreichbare Ortungsgenauigkeit durch verschiedene mathematische Verfahren eingegangen werden und zum anderen auf die Kombination mehrerer Sensoren. Außerdem wird noch einmal auf die betrachteten Testfälle eingegangen.

### 6.1 Vergleichsmöglichkeit und Einsatz mehrerer Klassifikatoren

Im Rahmen dieser Arbeit wurde ein Fingerprinting-Verfahren, das auf einer SVM basiert, zur Ortung der Beacons verwendet und ein Mehrheitsvotum ausschließlich mit verschiedenen Merkmalräumen durchgeführt. Nicht eindeutig ist jedoch, welches Fingerprinting-Verfahren am besten für Bluetooth-basierte Ortungsverfahren geeignet sind.

In der wissenschaftlichen Literatur werden häufig Bayes-Klassifikatoren oder KNN-Verfahren angewandt. Die Ergebnisse unterscheiden sich bisher nur geringfügig, allerdings wäre ein Vergleich dieser häufig genutzten Verfahren mit den hier vorgestellten SVM-Verfahren interessant. Ebenso wäre bei den Standard-Klassifikatoren interessant, ob auch bei diesen die Kombination mehrerer Merkmale ein besseres Ergebnis liefert, da auch hier bisher lediglich der naheliegende Mittelwert als Merkmal betrachtet wurde. Des Weiteren sind diese Klassifikatoren bereits in der JSAT-Bibliothek enthalten, sodass der Implementierungsaufwand vergleichsweise gering ausfällt.

Zusätzlich zu dem Vergleich mit Standard-Klassifikatoren, wäre ein Vergleich mit ungewöhnlicheren Klassifikatoren wünschenswert. Dazu zählen auf der empirischen Verteilungsfunktion basierte Verfahren, wie die Klassifikation über den Kolmogorov-Smirnow- oder den Cramer-von-Mises-Test. Die Idee hinter dieser Art von Verfahren ist, dass zunächst ein Fingerabdruck in Form einer empirischen Verteilungsfunktion erstellt wird.

Anschließend kann z.B. der Kolmogorov-Smirnow-Test mit Live-Daten durchgeführt werden. Dieser liefert ein Maß, das angibt, ob die live aufgenommenen Daten so verteilt sind, wie im Fingerabdruck. Je näher die Verteilung der Daten aus der Online-Phase der empirischen Verteilungsfunktion kommt, desto eher gehört das Signal in den entsprechenden Bereich.

Bisher wurde diese Art der Klassifikation nicht im Bereich der Innenraumlokalisierung eingesetzt, konnten jedoch im Rahmen von Signal-Modulationen vielversprechende Ergebnisse erzielen, da sie weniger anfällig für viele Formen von Signalschwankungen ist [WW10]. Insbesondere konnten die Klassifikation von Signalen verbessert werden, deren Störungen nicht normalverteilt waren, was bei Bluetooth-Beacons ebenfalls gilt.

Nachdem einige weitere Klassifikatoren untersucht und mit den aktuellen SVMs verglichen wurden, ist ebenso eine Erweiterung des Mehrheitsvotums denkbar. Dadurch würden nicht nur Entscheidungen, die auf unterschiedlichen Merkmalsräumen gründen, berücksichtigt, sondern völlig verschiedene Verfahren. Dadurch ist garantiert, dass die Entscheidungen der Verfahren unkorreliert sind.

Dies könnte einzelne Verfahrensschwächen besser ausgleichen als nur die Verwendung unterschiedlicher Merkmalsräume, da sich eine Korrelation nie vollständig vermeiden lässt. Die bisherigen Ergebnisse deuten darauf hin, dass sich die Ortungsgenauigkeit sowohl mit beweglichen Beacons als auch mit stationären Beacons noch einmal steigern ließe.

Des Weiteren könnten andere Ensemble-Techniken verwendet werden, um die Ergebnisse aus mehreren Klassifikatoren zu einem einzigen zusammenzuführen, da das im Rahmen dieser Ausarbeitung vorgestellte Mehrheitsvotum nur das einfachste dieser Verfahrensklasse ist. Eine erste Erweiterung könnte beispielsweise eine Gewichtung der einzelnen Ergebnisse wie in [Die00] sein.

## 6.2 Kombination mit anderen Sensoren

Da ein Smartphone mit einer entsprechenden App in ein Bluetooth-Beacon verwandelt werden kann, bietet sich die Möglichkeit, die Genauigkeit der Positionsbestimmung zu steigern, indem unterschiedliche Sensoren in die Lokalisierung einbezogen werden. Dafür können sowohl inertielle Sensoren wie Beschleunigungs- oder Erdmagnetfeldsensor dienen als auch die WLAN-Signalstärke. Ebenso können diese Sensoren hinzugezogen werden, wenn das Smartphone als Empfänger dient.

Dies hätte den Vorteil, dass versucht werden kann, nicht nur ein ungefähres Aufenthaltsgebiet zu bestimmen, sondern über Zusatzinformationen, die die Sensoren liefern, auf die tatsächliche Position geschlossen werden kann. Über die Verbindung mit Geometrieinformationen, die über Kartendaten ebenfalls vorhanden sind, ließen sich genauere Verfahren entwickeln, die nicht die Schwierigkeit aufweisen, dass die ermittelte Position einer Person physikalisch unmöglich ist oder die Trajektorien einer Person durch Wände führen.

Dabei handelt es sich um bisher häufig auftretende Hindernisse bei der Kombination mehrerer Sensoren zur Ortung. Ist daher bekannt, in welchem Gebiet sich eine Person befindet und mit welcher Geschwindigkeit sie sich bewegt, kann z.B. über einen Kalman-Filter direkt der Ort abgeleitet werden, wenn der Kalman-Filter auf einem entsprechenden Bewegungsmodell basiert. Dies wäre insbesondere für die Navigation von sich schneller bewegendem Objekten, wie zum Beispiel Gabelstapler, sinnvoll.

Durch die Kalman-Korrekturen kann allerdings der oben genannte Fall auftreten, dass die Position innerhalb von unerreichbaren Gebieten liegt, da durch eine genaue Positionsbestimmung Fehlerfortpflanzungen entstehen. Ein Abgleich mit vorhandenen Geometrieinformationen ist daher zur Vermeidung solcher Fälle hilfreich.

Dadurch könnten die Ortung über Bluetooth-Signale ein Referenzbereich liefern, in dem sich die zu lokalisierende Person theoretisch aufhalten müsste. Diese Position kann



danach mit den Bewegungsdaten und bisherigen Positionen verglichen werden. Dadurch würden sich gegenseitige Korrekturmöglichkeiten der Sensoren bieten, die in Verbindung mit Geometrieinformationen zu einer genaueren Ortung führen.

Insbesondere für die Untersuchung von Fußgängerdynamiken könnten sich durch die Kombination mehrerer Sensoren in Standard-Hardware neue günstige Möglichkeiten ergeben, um Personen innerhalb von Gebäuden zu verfolgen, wenn die Ortungsgenauigkeit auf unter einen Meter sinkt.

## 6.3 Standardisierung von Testszenarien

Der größte Schwierigkeit in der Bewertung von Verfahren zur Innenraumlokalisierung liegt in den Störungen, die während einer Live-Lokalisierung auf die Ortung einwirken. Um die Genauigkeit eines Verfahrens beurteilen zu können, müssen so viele Einflüsse wie möglich bekannt sein. Allerdings werden Einflüsse wie Material und Beschaffenheit der Wände in Ausarbeitungen nur selten erwähnt. Dadurch lassen sich Ergebnisse von unterschiedlichen Arbeiten schwer vergleichen. Auch die Geometrie ist in jeder Ausarbeitung eine andere, da jede Gruppe in ihren eigenen Laboren arbeitet.

In der Bildverarbeitung wurde dieses Problem umgangen, indem die Bilder an denen Verfahren getestet wurden, öffentlich zugänglich gemacht wurden. Dadurch ist Forschern die Möglichkeit gegeben, ihre Ergebnisse direkt mit den Ergebnissen anderer zu vergleichen, da sie das exakt gleiche Bild verwenden können. Eine solche Datenbank wird zum Beispiel von der Universität von Süd Kalifornien zur freien Verfügung gestellt[Uni81].

Damit auch die Ergebnisse von verschiedenen Verfahren zur Innenraum-Ortung verglichen werden können, sollte eine ähnliche Vorgehensweise etabliert werden. Dabei müssen jedoch deutlich mehr Parameter öffentlich gemacht werden als ein einzelnes Bild. So müssten mindestens Raumeometrie und Position von Sendern und Empfängern bekannt sein, sowie alle zugehörigen Rohdaten, die im optimalen Fall mit Zeitstempeln ausgestattet sind.

Zusätzlich sind jeweils ein Datensatz für die beiden Fingerprinting-Phasen zur Verfügung zu stellen. Dabei müssen jedoch zudem die realen Positionen während der Online-Phase bekannt sein, da sich sonst keine Vergleichsmöglichkeit bietet. Sollte eine Methode ohne Fingerprinting eingesetzt werden, könnte sich dieses Verfahren anhand der Daten aus der Online-Phase dennoch mit solchen Verfahren vergleichen lassen und denselben Datensatz benutzen.

Ein weiteres Problem der Vergleichbarkeit tritt auf, wenn verschiedene Sensoren mit in die Positionsbestimmung einbezogen werden. Verfahren, die nur auf Bluetooth oder WLAN basieren, ließen sich jedoch mit einem solchen Testdatensatz oder im sogar mehreren direkt miteinander vergleichen. Dadurch könnten sich Stärken und Schwächen einzelner Verfahren besser analysieren lassen und somit insgesamt zu einer Verbesserung der Ortungsgenauigkeit führen.

Eine Alternative zu Testdatensätzen wäre die Simulation verschiedener Umgebungen und Szenarien. Dazu müsste jedoch zunächst eine Simulation entwickelt werden, die möglichst viele Effekte der Funkwellenausbreitung berücksichtigt. Dadurch ließen sich

die Ergebnisse virtuell vergleichen. Dennoch wären weiterhin Versuche nötig, da eine solche Simulation die realen Daten nicht genau genug abbilden kann.

In jedem Fall sollte dafür gesorgt werden, dass die Ergebnisse verschiedener Verfahren miteinander verglichen werden können. In dieser Ausarbeitung konnten zum Beispiel gute Ergebnisse auf Raumbene erzielt werden. Interessant wäre jedoch zu wissen, ob dies dem Verfahren oder der Geometrie und den Wandeigenschaften zu verdanken ist, da beispielsweise nicht getestet werden konnte, wie sich die hier vorgestellten Verfahren bei leichten Spannbetonwänden verhalten.

# Literaturverzeichnis

- [App14] Apple. Getting started with ibeacon. <https://developer.apple.com/ibeacon/Getting-Started-with-iBeacon.pdf>, 2014. [Online; zugegriffen 11. August 2016].
- [Arg16] Argenox. ble advertising channel. <http://www.argenox.com/wp-content/uploads/ble-advertising-channels-spectrum.png>, 2016. [Online; zugegriffen 6. August 2016].
- [AY14] Khalid AA Abakar and Chongwen Yua. Performance of svm based on puk kernel in comparison to svm based on rbf kernel in prediction of yarn tenacity. *Indian Journal of Fibre & Textile Research*, 39:55–59, 2014.
- [BaS15] BaSiGo. Bausteine für die Sicherheit von Großveranstaltungen. <http://www.basigo.de/>, 2015. [Online; zugegriffen 12. August 2016].
- [Bol15] Maik Boltes. *Automatische Erfassung präziser Trajektorien in Personenströmen hoher Dichte*. Dr., Universität zu Köln, Jülich, 2015. Universität zu Köln, Diss., 2014.
- [CSE04] Javier García Castaño, Michael Svensson, and M Ekstrom. Local positioning for wireless sensors based on bluetooth<sup>TM</sup>. In *Radio and Wireless Conference, 2004 IEEE*, pages 195–198. IEEE, 2004.
- [Die00] Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.
- [FDW04] Gunter Fischer, Burkhard Dietrich, and Frank Winkler. Bluetooth indoor localization system. In *Proceedings of the 1st Workshop on Positioning, Navigation and Communication*, 2004.
- [FH14] Ramsey Faragher and R Harle. An analysis of the accuracy of bluetooth low energy for indoor positioning applications. In *Proceedings of the 27th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2014), Tampa, FL, USA*, volume 812, 2014.
- [Fou16] Raspberry Pi Foundation. <https://www.raspberrypi.org>, 2016. [Online; zugegriffen 13. September 2016].
- [JuP16] JuPedSim. Juelich Pedestrian Simulator. <http://www.jupedsim.org>, 2016. [Online; zugegriffen 11. August 2016].

- [KK04] Kamol Kaemarungsi and Prashant Krishnamurthy. Properties of indoor received signal strength for wlan location fingerprinting. In *Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004. The First Annual International Conference on*, pages 14–23. IEEE, 2004.
- [KMU<sup>+</sup>14] Shinsuke Kajioka, Tomoya Mori, Takahiro Uchiya, Ichi Takumi, and Hiroshi Matsuo. Experiment of indoor position presumption based on rssi of bluetooth le beacon. In *2014 IEEE 3rd Global Conference on Consumer Electronics (GCCE)*, pages 337–339. IEEE, 2014.
- [Mey09] Steffen Meyer. Signalstärke-basierte Lokalisierung – Basis für ortsabhängige Dienste, 2009.
- [Mis13] Sandeep Mistry. noble. <https://github.com/sandeepmistry/noble>, 2013. [Online; zugegriffen 20. August 2016].
- [Mul16] Multicherry. Raspberry Pi B+. <https://commons.wikimedia.org/w/index.php?curid=38504663>, 2016. [Online; zugegriffen 6. August 2016].
- [RIM16] RIMEA. Richtlinie für Mikroskopische Entfluchtungs-Analysen. <http://www.rimea.de>, 2016. [Online; zugegriffen 11. August 2016].
- [SHRB11] Fazli Subhan, Halabi Hasbullah, Azat Rozyyev, and Sheikh Tahir Bakhsh. Indoor positioning in bluetooth networks using fingerprinting and lateration approach. In *2011 International Conference on Information Science and Applications*, pages 1–9. IEEE, 2011.
- [SIG16] SIG. Bluetooth Special Interest Group. <https://www.bluetooth.com>, 2016. [Online; zugegriffen 11. August 2016].
- [The14] The Android Open Source Project. ScanManager.java. [http://androidxref.com/5.0.0\\_r2/xref/packages/apps/Bluetooth/src/com/android/bluetooth/gatt/ScanManager.java](http://androidxref.com/5.0.0_r2/xref/packages/apps/Bluetooth/src/com/android/bluetooth/gatt/ScanManager.java), 2014. [Online; zugegriffen 11. August 2016].
- [Tre15] Eugen Trebunski. Innenraum-lokalisierung mittels Bluetooth-Smart-Beacons. Master’s thesis, Fachhochschule Aachen, Campus Jülich, 2015.
- [Uni81] Signal and Image Processing Institute University of Southern California. The usc-sipi image database. <http://sipi.usc.edu/database/>, seit 1981. [Online; zugegriffen 20. September 2016].
- [War14] Adam Warski. How do ibeacons work? <http://www.warski.org/blog/2014/01/how-ibeacons-work/>, 2014. [Online; zugegriffen 14. August 2016].
- [Wik16a] Wikipedia. eCall. <https://de.wikipedia.org/wiki/ECall>, 2016. [Online; zugegriffen 7. August 2016].

- [Wik16b] Wikipedia. Global Positioning System. [https://de.wikipedia.org/wiki/Global\\_Positioning\\_System](https://de.wikipedia.org/wiki/Global_Positioning_System), 2016. [Online; zugegriffen 6. August 2016].
- [WW10] Fanggang Wang and Xiaodong Wang. Fast and robust modulation classification via kolmogorov-smirnov test. *IEEE Transactions on Communications*, 58(8):2324–2332, 2010.
- [You13] David G. Young. android-beacon-library. <https://github.com/AltBeacon/android-beacon-library>, 2013. [Online; zugegriffen 11. August 2016].